Khaled Mellouli (Ed.)

LNAI 4724

Symbolic and Quantitative Approaches to Reasoning with Uncertainty

9th European Conference, ECSQARU 2007
Hammamet, Tunisia, October/November 2007
Proceedings

Lecture Notes in Artificial Intelligence 4724

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Khaled Mellouli (Ed.)

Symbolic and Quantitative Approaches to Reasoning with Uncertainty

9th European Conference, ECSQARU 2007
Hammamet, Tunisia, October 31 - November 2, 2007
Proceedings

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editor

Khaled Mellouli
LARODEC
Institut Supérieur de Gestion de Tunis
41 Avenue de la liberté
2000 Le Bardo, Tunisie
E-mail: khaled.mellouli@ihec.rnu.tn

Library of Congress Control Number: 2007935607

CR Subject Classification (1998): I.2, F.4.1

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-75255-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-75255-4 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12166039 06/3180 5 4 3 2 1 0

Preface

These are the proceedings of the Ninth European Conference on Symbolic and Qualitative Approaches to Reasoning with Uncertainty, ECSQARU 2007, held in Hammamet (Tunisia), October 31, November 1–2 2007. The biannual ECSQARU conferences are a major forum for advances in the theory and practice of reasoning under uncertainty. The first ECSQARU conference was held in Marseille (1991), and since then it has been held in Granada (1993), Fribourg (1995), Bonn (1997), London (1999), Toulouse (2001), Aalborg (2003) and Barcelona (2005).

The papers gathered in this volume were selected out of 120 submissions from 25 countries, after a rigorous reviewing process by three Program Committee members and a restricted Program Committee. In addition to the regular presentations, the technical program of ECSQARU 2007 also included invited lectures by three outstanding researches: Thierry Denoeux (Pattern Recognition and Information Fusion Using Belief Functions: Some Recent Developments), Salem Benferhat (Qualitative Models for Reasoning Under Uncertainty: From Non-monotonicity and Causality to Applications) and Anthony Hunter (Elements of Argumentation).

A special dedication to the memory of Philippe Smets, who collaborated with our team for several years and who gave full support to the organization of ECSQARU in Tunisia. In his memory we have organized a permanent special session grouping an overview of his main contributions in the artificial intelligence field.

Finally, I would like to thank the members of the Program Committee and all the additional referees for their valuable work. I also want to express my acknowledgment to all of my colleagues of the Organizing Committee for their support in making this conference successful.

July 2007 Khaled Mellouli

Organization

ECSQARU 2007 was organized by Laboratoire de Recherche Operationnelle, de Decision et de Controle de Processus (LARODEC) and Tunisian Management Science Society (TMSS) (Institut Superieur de Gestion Tunis, Tunisia).

Executive Committee

Conference Chair	Khaled Mellouli (IHEC, Tunisia)
Organizing Committee	Nahla Ben Amor (ISG, Tunisia)
	Amel Ben Yaghlane (ISG, Tunisia)
	Boutheina Ben Yaghlane (IHEC, Tunisia)
	Zied Elouedi (ISG, Tunisia)
	Hajer Esseghir (ISG, Tunisia)
	Ilyes Jenhani (ISG, Tunisia)
	Fedia Khalfallah (FSEGT, Tunisia)

Program Committee

Foued Ben Abdelaziz (Tunisia)
Nahla Ben Amor (Tunisia)
Salem Benferhat (France)
Boutheina Ben Yaghlane (Tunisia)
Isabelle Bloch (France)
Philippe Besnard (France)
Gerd Brewka (Germany)
Luis M. de Campos (Spain)
Carlos Chesevar (Spain)
Giulianella Coletti (Italy)
Fabio Cozman (Brazil)
Adnan Darwiche (USA)
James P. Delgrande (Canada)
Thierry Denoeux (France)
Habiba Drias (Algeria)
Marek Druzdzel (USA)
Didier Dubois (France)
Abdelkader El Kamel (France)
Zied Elouedi (Tunisia)
Hlne Fargier (France)
Linda van der Gaag (Netherlands)
Hector Geffner (Spain)
Angelo Gilio (Italy)
Lluis Godo (Spain)
Michel Grabisch (France)
Petr Hjek (Czech Republic)
Andreas Herzig (France)
Eyke Hellermeier (Germany)
Anthony Hunter (UK)
Audun Jsang (Australia)
Gabriele Kern-Isberner (Germany)
Rudolf Kruse (Germany)
Jrme Lang (France)
Jonathan Lawry (UK)
Pedro Larraaga (Spain)
Churn-Jung Liau (Taiwan)
Weiru Liu (UK)
Thomas Lukasiewicz (Italy)
Pierre Marquis (France)
Serafn Moral (Spain)
Thomas Nielsen (Denmark)
Kristian Olesen (Denmark)
Ewa Orlowska (Poland)
Odile Papini (France)

Simon Parsons (USA)
Lus Moniz Pereira (Portugal)
Ramon Pino-Prez (Venezuela)
David Poole (Canada)
Henri Prade (France)
Mohamed Ramdani (France)
Abdelwahab Rebai (Tunisia)
Maria Rifqi (France)
Alessandro Saffiotti (Sweden)
Sandra Sandri (Spain)
Torsten Schaub (Germany)
Romano Scozzafava (Italy)
Glenn Shafer (USA)
Prakash P. Shenoy (USA)
Guillermo Simari (Argentina)
Claudio Sossai (Italy)
Milan Studeny (Czech Republic)
Leon van der Torre (Netherlands)
Emil Weydert (Luxembourg)
Mary-Anne Williams (Australia)
Nevin L. Zhang (HK, China)

Additional Reviewers

Carol Adam
Teresa Alsinet
Guillaume Aucher
Gaetano Chemello
Tao Chen
Sylvie Coste-Marquis
Patricia Everaere
Rim Faiz
Andrea Formesani
Benoit Gaudou
Martin Gebser
Joanna Golinska-Pillarek
Janusz Granat
Juan Huete
Robert Jeansoulin
Jaber Jemai
Mohamed Kissi
Sebastien Konieczny
Ivan Kramosil
Sylvain Lagrue
Helge Langseth
Dominique Longin
Tsai-Ching Lu
Jianbing Ma
Sehl Mellouli
Robert Mercer
Matthias Steinbrecher
Andzej Szalas
Barbara Vantaggi
Ivan Varzinczak
Yi Wang
Eric Wurbel
Safa Yahi
Adam Zagorecki

Sponsoring Institutions

Tunisian Management Science Society (TMSS)
Institut Superieur de Gestion Tunis (ISG, Tunis)
Institut des Hautes Etudes Commerciales de Carthage (IHEC, Carthage)

Table of Contents

Invited Talks

Causal Networks

Belief Revision and Inconsistency Handling

Logics Under Uncertainty

Argumentation Systems

Belief Functions

Learning and Classification

Bayesian Networks and Probabilistic Reasoning

Reasoning About Preferences

Reasoning and Decision Making Under Uncertainty

Game Theory

Fuzzy Sets and Fuzzy Logic

Possibility Theory

Applications

Pattern Recognition and Information Fusion Using Belief Functions: Some Recent Developments

Thierry Denœux

HEUDIASYC, UTC, CNRS
Centre de Recherche de Royallieu
BP 20529, F-60205 Compiègne, France

The Transferable Belief Model (TBM) is a general framework for reasoning with uncertainty using belief functions [8]. Of particular interest is the General Bayesian Theorem (GBT), an extension of Bayes's theorem in which probability measures are replaced by belief functions, and no prior knowledge is assumed [7,6].

Until recently, applications of the GBT have been fairly limited, mainly because of lack of methods for constructing belief functions from observation data. The availability of such methods [4,2,1] as well as new combination rules for merging partially overlapping items of evidence [5] now extend the applicability of the TBM to a wider class of statistical pattern recognition and information fusion tasks. These recent developments will be reviewed, and applications to various problems such as novelty detection [3] and partially supervised learning using mixture models will be discussed.

This talk is mostly self-contained. Relevant material (papers, slides) can be found at `http://www.hds.utc.fr/~tdenoeux`.

References

1. Aregui, A., Denœux, T.: Consonant belief function induced by a confidence set of pignistic probabilities. In: Proceedings of ECSQARU '2007, Hammamet, Tunisia (to appear)
2. Aregui, A., Denœux, T.: Constructing predictive belief functions from continuous sample data using confidence bands. In: De Cooman, G., Vejnarová, J., Zaffalon, M. (eds.) Proceedings of the Fifth International Symposium on Imprecise Probability: Theories and Applications (ISIPTA '07), July 2007, Prague, Czech Republic, pp. 11–20 (2007)
3. Aregui, A., Denœux, T.: Fusion of one-class classifiers in the belief function framework. In: Proceedings of the 10th Int. Conf. on Information Fusion, Canada (July 2007)
4. Denœux, T.: Constructing belief functions from sample data using multinomial confidence regions. International Journal of Approximate Reasoning 42(3), 228–252 (2006)
5. Denœux, T.: Conjunctive and disjunctive combination of belief functions induced by non distinct bodies of evidence. Artificial Intelligence 2007 (in press)
6. Denœux, T., Smets, P.: Classification using belief functions: the relationship between the case-based and model-based approaches. IEEE Transactions on Systems, Man and Cybernetics B 36(6), 1395–1406 (2006)
7. Smets, P.: Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem. International Journal of Approximate Reasoning 9, 1–35 (1993)
8. Smets, P., Kennes, R.: The Transferable Belief Model. Artificial Intelligence 66, 191–243 (1994)

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, p. 1, 2007.

Causality and Dynamics of Beliefs in Qualitative Uncertainty Frameworks

Salem Benferhat

CRIL-CNRS, université d'Artois. Rue Jean Souvraz. 62307 Lens Cedex. France
benferhat@cril.univ-artois.fr

Causality and belief changes play an important role in many applications. Recently, Pearl [6] has proposed approaches based on probability theory using causal graphs to give formal semantics to the notion of interventions. From representational point of view, interventions are distinguished from observations using the concept of the "do" operator [4]. From reasoning point of view, handling interventions consists in "ignoring" the effects of all direct (and undirected) causes related to the variable of interest.

This talk discusses causality in qualitative uncertainty frameworks, such as ordinal conditional functions frameworks [7] or possibility theory frameworks [2], with a particular focus on graphical models. We will also discuss the relationships between qualitative handling of causality and belief changes (belief revision and updating) [3], [5].

A first natural question is whethere the different equivalent representations of interventions in probabilistic networks are still valid in qualitative settings. We argue that this is the case since basically the main changes between probabilistic networks and probabilistic causal networks concerns the graphical structure.

The second question that is addressed concerns computational issues. Namely, we discuss whethere the handling of interventions, especially sequences of observations and interventions, has consequences on propagation algorithms. In particular, we are interested to know whethere one can reuse exisiting algorithms and what are extra costs induced by handling interventions.

The last and most important question concerns new issues that are arised in belief changes when handling interventions. We first point out that the order in which observations and interventions are introduced is very important. Then we argue that if one wants to respect this order then we need to change the structure of the graphical model after each observation and intervetion.

Some part of this work has been jointly done with S. Smaoui [1].

Acknowlegment. This work is supported by a french national project ANR Blanc Micrac.

References

1. Benferhat, S., Smaoui, S.: Possibilistic causal networks for handling interventions: A new propagation algorithm. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-2007), Vancouver, British Columbia, pp. 373–378 (2007)

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 2–3, 2007.

2. Dubois, D., Prade, H.: Possibility theory: An approach to computerized, Processing of uncertainty. Plenium Press, New York (1988)
3. Gärdenfors, P.: Knowledge in Flux: Modeling the Dynamics of Epistemic States. Bradford Books. MIT Press, Cambridge (1988)
4. Goldszmidt, M., Pearl, J.: Rank-based systems: A simple approach to belief revision, belief update, and reasoning about evidence and actions. In: Proceeding of the Third International Conference (KR'92), pp. 661–672. Kaufmann, San Francisco (1992)
5. Katsuno, H., Mendelzon, A.: Propositional Knowledge Base Revision and Minimal Change. Artificial Intelligence 52, 263–294 (1991)
6. Pearl, J.: Causality: models, reasoning, and inference. Cambridge University Press, New York (2000)
7. Spohn, W.: Ordinal conditional functions: a dynamic theory of epistemic states causation in decision. In: Harper, W., Skyrms, B. (eds.) Belief Changes and Statistics, pp. 105–134 (1988)

Elements of Argumentation

Anthony Hunter*

Department of Computer Science
University College London
Gower Street, London WC1E 6BT, UK

Abstract. Logic-based formalizations of argumentation, that take pros and cons for some claim into account, have been extensively studied, and some basic principles have been established (for reviews see [1–3]). These formalizations assume a set of formulae and then exhaustively lay out arguments and counterarguments, where a counterargument either rebuts (i.e. negates the claim of the argument) or undercuts (i.e. negates the support of the argument). Recently attempts have been made to refine these formalizations by using techniques for selecting the more appropriate arguments and counterarguments by taking into account intrinsic factors (such as the degree of inconsistency between an argument and its counterarguments) and extrinsic factors (such as the impact of particular arguments on the audience and the beliefs of the audience). In this presentation, we consider the need to take intrinsic and extrinsic factors into account, and then consider ways that this can be done in logic in order to refine existing logic-based approaches to argumentation. These refinements offer interesting options for formalizations that may better capture practical argumentation for intelligent agents [3].

References

1. Chesñevar, C., Maguitman, A., Loui, R.: Logicalmodelsofargument. ACMComputingSurveys 32(4), 343–387 (2000)
2. Prakken, H., Vreeswijk, G.: Logicalsystemsfordefeasibleargumentation. In: Gabbay, D., Guenthner, F. (eds.) HandbookofPhilosophicalLogic, 2nd edn., vol. 4, pp. 219–318. Kluwer, Dordrecht (2002)
3. Besnard, P., Hunter, A.: ElementsofArgumentation. MIT Press, Cambridge (2008)

* This is work jointly undertaken with Philippe Besnard (IRIT, Université Paul Sabatier, 118 route de Narbonne, 31062 Toulouse, France).

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, p. 4, 2007.

Causal Graphical Models with Latent Variables: Learning and Inference

Stijn Meganck[1], Philippe Leray[2], and Bernard Manderick[1]

[1] CoMo, Vrije Universiteit Brussel, 1050 Brussels, Belgium
smeganck@vub.ac.be
http://como.vub.ac.be/Members/stijn/stijn.html
[2] LITIS EA 4051, INSA Rouen, St. Etienne du Rouvray, France

Abstract. Several paradigms exist for modeling causal graphical models for discrete variables that can handle latent variables without explicitly modeling them quantitatively. Applying them to a problem domain consists of different steps: structure learning, parameter learning and using them for probabilistic or causal inference. We discuss two well-known formalisms, namely semi-Markovian causal models and maximal ancestral graphs and indicate their strengths and limitations. Previously an algorithm has been constructed that by combining elements from both techniques allows to learn a semi-Markovian causal models from a mixture of observational and experimental data. The goal of this paper is to recapitulate the integral learning process from observational and experimental data and to demonstrate how different types of inference can be performed efficiently in the learned models. We will do this by proposing an alternative representation for semi-Markovian causal models.

1 Introduction

This paper discusses causal graphical models for discrete variables that can handle latent variables without explicitly modeling them quantitatively. In the *uncertainty in artificial intelligence* area there exist several paradigms for such problem domains. Two of them are *semi-Markovian causal models* and *maximal ancestral graphs*. Applying these techniques to a problem domain consists of several steps, typically: structure learning from observational and experimental data, parameter learning, probabilistic inference, and, quantitative causal inference.

We will discuss the fact that each of the existing approaches for causal modeling with latent variables only focuses on one or a few of all the steps involved in the process of modeling a problem.

Semi-Markovian causal models (SMCMs) are an approach developed by Tian and Pearl [1,2]. They are specifically suited for performing quantitative causal inference in the presence of latent variables. However, at this time no efficient parametrisation of such models is provided and there is no algorithm for performing efficient probabilistic inference. Furthermore there are no techniques to learn these models from data issued from observations, experiments or both.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 5–16, 2007.

Maximal ancestral graphs (MAGs) [3] are specifically suited for structure learning in the presence of latent variables from observational data. However, the techniques only learn up to Markov equivalence and provide no clues on which additional experiments to perform in order to obtain the fully oriented causal graph. See [4,5] for that type of results for Bayesian networks without latent variables. Furthermore, as of yet no parametrisation for discrete variables is provided for MAGs and no techniques for probabilistic inference have been developed. There is some work on algorithms for causal inference, but it is restricted to causal inference quantities that are the same for an entire Markov equivalence class of MAGs [6,7].

We have chosen to use SMCMs as a final representation in our work, because they are the only formalism that allows to perform causal inference while fully taking into account the influence of latent variables. In previous work [8] we combined the existing techniques to learn MAGs with newly developed methods to provide an integral approach that uses both observational data and experiments in order to learn fully oriented semi-Markovian causal models.

In this paper we develop an alternative representation for the probability distribution represented by a SMCM, together with a parametrisation for this representation, where the parameters can be learned from data with classical techniques. Finally, we discuss how probabilistic and quantitative causal inference can be performed in these models with the help of the alternative representation and its associated parametrisation.

The next section introduces some notations and definitions and we discuss causal models with latent variables. After that we discuss structure learning for those models and in the next section we introduce techniques for learning a SMCM with the help of experiments. Then we introduce a new representation for SMCMs that can easily be parametrised. We also show how both probabilistic and causal inference can be performed with the help of this new representation.

2 Preliminaries

We start this section by introducing basic notations necessary for the understanding of the rest of this paper. Then we will discuss classical probabilistic Bayesian networks followed by causal Bayesian networks. Finally we handle the difference between probabilistic and causal inference, or observation vs. manipulation.

2.1 Notations

In this work uppercase letters are used to represent variables or sets of variables, i.e. $V = \{V_1, \ldots, V_n\}$, while corresponding lowercase letters are used to represent their instantiations, i.e. v_1, v_2 and v is an instantiation of all v_i. $P(V_i)$ is used to denote the probability distribution over all possible values of variable V_i, while $P(V_i = v_i)$ is used to denote the probability of the instantiation of variable V_i to value v_i. Usually, $P(v_i)$ is used as an abbreviation of $P(V_i = v_i)$.

The operators $Pa(V_i), Anc(V_i), Ne(V_i)$ denote the observable parents, ancestors and neighbors respectively of variable V_i in a graph and $Pa(v_i)$ represents the values of the parents of V_i. If $V_i \leftrightarrow V_j$ appears in a graph then we say that they are spouses, i.e. $V_i \in Sp(V_j)$ and vice versa.

When two variables V_i, V_j are independent we denote it by $(V_i \perp\!\!\!\perp V_j)$, when they are dependent by $(V_i \not\perp\!\!\!\perp V_j)$.

2.2 Definitions

Both techniques are an extension of causal Bayesian networks for modeling systems without latent variables.

Definition 1. *A **Causal Bayesian Network** is a triple $\langle V, G, P(v_i|Pa(v_i))\rangle$, with:*

- *$V = \{V_1, \ldots, V_n\}$, a set of observable discrete random variables*
- *a directed acyclic graph (DAG) G, where each node represents a variable from V*
- *parameters: conditional probability distributions (CPD) $P(v_i|Pa(v_i))$ of each variable V_i from V conditional on its parents in the graph G.*
- *Furthermore, the directed edges in G represent autonomous causal relations between the corresponding variables.*

The interpretation of directed edges is different from a classical BN, where the arrows only represent a probabilistic dependency, and not necessarily a causal one.

This means that in a CBN, each CPD $P(v_i|Pa(v_i))$ represents a stochastic assignment process by which the values of V_i are chosen in response to the values of $Pa(V_i)$ in the underlying domain. This is an approximation of how events are physically related with their effects in the domain that is being modeled. For such an assignment process to be autonomous means that it must stay invariant under variations in the processes governing other variables [1].

In the above we made the assumption of *causal sufficiency*, i.e. that for every variable of the domain that is a common cause, observational data can be obtained in order to learn the structure of the graph and the CPDs. Often this assumption is not realistic, as it is not uncommon that a subset of all the variables in the domain is never observed. We refer to such a variable as a *latent* variable.

The central graphical modeling representation that we use are the semi-Markovian causal models. They were first used by Pearl [1], and Pearl and Tian [2] have developed causal inference algorithms for them.

Definition 2. *A **Semi-Markovian Causal Model** (SMCM) is an acyclic causal graph G with both directed and bi-directed edges. The nodes in the graph represent observable variables $V = \{V_1, \ldots, V_n\}$ and the bi-directed edges implicitly represent latent variables $L = \{L_1, \ldots, L_{n'}\}$.*

See Figure 1(b) for an example SMCM representing the underlying DAG in (a).

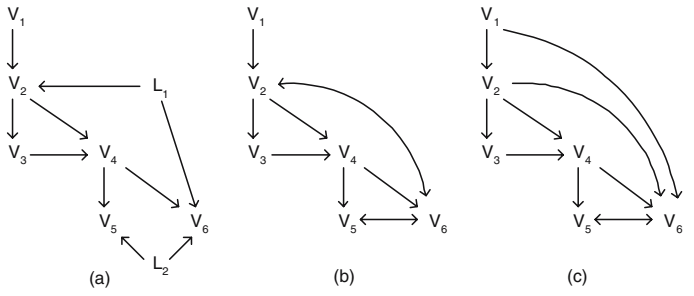


Fig. 1. (a) A problem domain represented by a causal DAG model with observable and latent variables. (b) A semi-Markovian causal model representation of (a). (c) A maximal ancestral graph representation of (a).

Maximal ancestral graphs are another approach to modeling with latent variables [3]. The main research focus in that area lies on learning the structure of these models and on representing exactly all the independences between the observable variables of the underlying DAG.

Ancestral graphs (AGs) are graphs that are complete under marginalisation and conditioning. We will only discuss AGs without conditioning as is commonly done in recent work [9,10,11].

Definition 3. *An **ancestral graph** without conditioning is a graph with no directed cycle containing directed* $\rightarrow$ *and bi-directed* $\leftrightarrow$ *edges, such that there is no bi-directed edge between two variables that are connected by a directed path.*

Definition 4. *An ancestral graph is said to be a **maximal ancestral graph** if, for every pair of non-adjacent nodes* V_i, V_j *there exists a set* Z *such that* V_i *and* V_j *are d-separated given* Z*.*

See Figure 1(c) for an example MAG representing the underlying DAG in (a) and corresponding to the SMCM in (b).

Definition 5. *Let* $[G]$ *be the Markov equivalence class for an arbitrary MAG* G*. The **complete partial ancestral graph** (CPAG) for* $[G]$*,* P_G*, is a graph with possibly the following edges* $\rightarrow, \leftrightarrow, o\!\!-\!\!o, o\!\!\rightarrow$*, such that*

1. P_G *has the same adjacencies as* G *(and hence any member of* $[G]$*) does;*
2. *A mark of arrowhead* $(>)$ *is in* P_G *if and only if it is invariant in* $[G]$*; and*
3. *A mark of tail* $(-)$ *is in* P_G *if and only if it is invariant in* $[G]$*.*
4. *A mark of* (o) *is in* P_G *if not all members in* $[G]$ *have the same mark.*

See Figure 2 for the corresponding CPAG for the MAG shown in Figure 1(c).

3 Structure Learning with Latent Variables

Just as learning a graphical model in general, learning a model with latent variables consists of two parts: structure learning and parameter learning. Both can

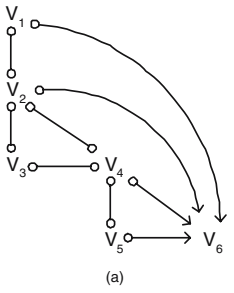


Fig. 2. The CPAG corresponding to the MAG in Figure 1(c)

be done using data, expert knowledge and/or experiments. In this section we discuss structure learning and we differentiate between learning from observational and experimental data.

3.1 Learning from Observational Data

In the literature no algorithm for learning the structure of an SMCM exists. In order to learn MAGs from observational data a constraint based learning algorithm has been developed. It is called the Fast Causal Inference (FCI) algorithm [12] and it uses conditional independence relations found between observable variables to learn a structure. Recently this result has been extended with the complete tail augmentation rules introduced in [13]. The results of this algorithm is a complete partial ancestral graph (CPAG), representing the Markov equivalence class of MAGs consistent with the data.

In a CPAG the directed edges have to be interpreted as representing ancestral relations instead of immediate causal relations. More precisely, this means that there is a directed edge from V_i to V_j if V_i is an ancestor of V_j in the underlying DAG and there is no subset of observable variables D such that $(V_i \perp\!\!\!\perp V_j | D)$. This does not necessarily mean that V_i has an immediate causal influence on V_j, it may also be a result of an inducing path between V_i and V_j. An inducing path between V_i and V_j is a path that can not be blocked by any subset of variables, the official definition is given below:

Definition 6. *An **inducing path** is a path in a graph such that each observable non-endpoint node is a collider, and an ancestor of at least one of the endpoints.*

A consequence of these properties of MAGs and CPAGs is that they are not very suited for general causal inference, since the immediate causal parents of each observable variable are not available and this information is needed to perform the calculations. As we want to learn models that can perform causal inference, we will discuss how to transform a CPAG into a SMCM next and hence introduce a learning algorithm for SMCMs using both observational and experimental data.

3.2 Learning from Experimental Data

As mentioned above, the result of current state-of-the-art techniques that learn models with implicit latent variables from observational data is a CPAG. This is a representative of the Markov equivalence class of MAGs. Any MAG in that class will be able to represent the same JPD over the observable variables, but not all those MAGs will have all edges with a correct causal orientation.

Furthermore in MAGs the directed edges do not necessarily have an immediate causal meaning as in CBNs or SMCMs, instead they have an ancestral meaning. If it is your goal to perform causal inference, you will need to know the immediate parents to be able to reason about all causal queries.

MAGs are maximal, thus every missing edge must represent a conditional independence. In the case that there is an inducing path between two variables and no edge in the underlying DAG, the result of the current learning algorithms will be to add an edge between the variables. Again, although these type of edges give the only correct representation of the conditional independence relations in the domain, they do not represent an immediate causal relation (if the inducing edge is directed) or a real latent common cause (if the inducing edge is bi-directed). Because of this they could interfere with causal inference algorithms, therefore we would like to identify and remove these type of edges.

To recapitulate, the goal of techniques aiming at transforming a CPAG must be twofold:

- finding the correct causal orientation of edges that are not completely specified by the CPAG ($o\!\!\rightarrow$ or $o\!\!-\!\!o$), and,
- removing edges due to inducing paths.

For the details of the learning algorithm we refer to [14] and [8]. For the remainder of the paper we will focus on constructing an alternative representation for SMCMs in order to perform inference.

4 Parametrisation of SMCMs

In his work on causal inference, Tian provides an algorithm for performing causal inference given knowledge of the structure of an SMCM and the joint probability distribution (JPD) over the observable variables. However, a parametrisation to efficiently store the JPD over the observables is not provided.

We start this section by discussing the factorisation for SMCMs introduced in [2]. From that result we derive an additional representation for SMCMs and a parametrisation of that representation that facilitates probabilistic and causal inference. We will also discuss how these parameters can be learned from data.

4.1 Factorising with Latent Variables

Consider an underlying DAG with observable variables $V = \{V_1, \ldots, V_n\}$ and latent variables $L = \{L_1, \ldots, L_{n'}\}$. Then the joint probability distribution can be written as the following mixture of products:

$$P(v) = \sum_{\{l_k | L_k \in L\}} \prod_{V_i \in V} P(v_i | Pa(v_i), LPa(v_i)) \prod_{L_j \in L} P(l_j), \tag{1}$$

where $LPa(v_i)$ are the latent parents of variable V_i.

Remember that in a SMCM the latent variables are implicitly represented by bi-directed edges, then consider the following definition.

Definition 7. *In a SMCM, the set of observable variables can be partitioned into disjoint groups by assigning two variables to the same group iff they are connected by a bi-directed path. We call such a group a* ***c-component*** *(from "confounded component") [2].*

E.g. in Figure 1(b) variables V_2, V_5, V_6 belong to the same c-component. Then it can be readily seen that c-components and their associated latent variables form respective partitions of the observable and latent variables. Let $Q[S_i]$ denote the contribution of a c-component with observable variables $S_i \subset V$ to the mixture of products in equation 1. Then we can rewrite the JPD as follows: $P(v) = \prod_{i \in \{1,\ldots,k\}} Q[S_i]$.

Finally, in [2] it is shown that each $Q[S]$ could be calculated as follows. Let $V_{o_1} < \ldots < V_{o_n}$ be a topological order over V, and let $V^{(i)} = \{V_{o_1}, \ldots, V_{o_i}\}$, $i = 1, \ldots, n$ and $V^{(0)} = \emptyset$.

$$Q[S] = \prod_{V_i \in S} P(v_i | (T_i \cup Pa(T_i)) \backslash \{V_i\}) \tag{2}$$

where T_i is the c-component of the SMCM G reduced to variables $V^{(i)}$, that contains V_i. The SMCM G reduced to a set of variables $V' \subset V$ is the graph obtained by removing all variables $V \backslash V'$ from the graph and the edges that are connected to them.

In the rest of this section we will develop a method for deriving a DAG from a SMCM. We will show that the classical factorisation $\prod P(v_i | Pa(v_i))$ associated with this DAG, is the same as the one that is associated with the SMCM as above.

4.2 Parametrised Representation

Here we first introduce an additional representation for SMCMs, then we show how it can be parametrised and finally, we discuss how this new representation could be optimised.

PR-Representation. Consider $V_{o_1} < \ldots < V_{o_n}$ to be a topological order O over the observable variables V, and let $V^{(i)} = \{V_{o_1}, \ldots, V_{o_i}\}$, $i = 1, \ldots, n$ and $V^{(0)} = \emptyset$. Then Table 1 shows how the parametrised (PR-) representation can be obtained from the original SMCM structure.

What happens is that each variable becomes a child of the variables it would condition on in the calculation of the contribution of its c-component as in Equation (2).

Table 1. Obtaining the parametrised representation from a SMCM

Given a SMCM G and a topological order O, the PR-representation has these properties:
1. The nodes are V, the observable variables of the SMCM.
2. The directed edges that are present in the SMCM are also present in the PR-representation.
3. The bi-directed edges in the SMCM are replaced by a number of directed edges in the following way: Add an edge from node V_i to node V_j iff: a) $V_i \in (T_j \cup Pa(T_j))$, where T_j is the c-component of G reduced to variables $V^{(j)}$ that contains V_j, b) except if there was already an edge between nodes V_i and V_j.

In Figure 3(a), the PR-representation of the SMCM in Figure 1(a) can be seen. The topological order that was used here is $V_1 < V_2 < V_3 < V_4 < V_5 < V_6$ and the directed edges that have been added are $V_1 \rightarrow V_5$, $V_2 \rightarrow V_5$, $V_1 \rightarrow V_6$, $V_2 \rightarrow V_6$, and, $V_5 \rightarrow V_6$.

The resulting DAG is an I-map [15], over the observable variables of the independence model represented by the SMCM. This means that all the independences that can be derived from the new graph must also be present in the JPD over the observable variables. This property can be more formally stated as the following theorem.

Theorem 1. *The PR-representation PR derived from a SMCM S is an* ***I-map*** *of that SMCM.*

Proof. Proving that PR is an I-map of S amounts to proving that all independences represented in PR (A) imply an independence in S (B), or $A \Rightarrow B$. We will prove that assuming both A and $\neg B$ leads to a contradiction.

Assumption $\neg B$: consider that two observable variables X and Y are dependent in the SMCM S conditional on some (possible empty) set of observable variables Z: $X \not\perp\!\!\!\perp_S Y | Z$.

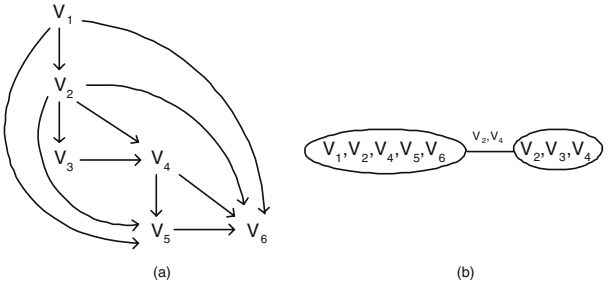


Fig. 3. (a) The PR-representation applied to the SMCM of Figure 1(b). (b) Junction tree representation of the DAG in (a).

Assumption A: consider that X and Y are independent in PR conditional on Z: $X \perp\!\!\!\perp_{PR} Y|Z$.

Then based on $X \not\perp\!\!\!\perp_S Y|Z$ we can discriminate two general cases:

1. $\exists$ a path C in S connecting variables X and Y that contains no colliders and no elements of Z.
2. $\exists$ a path C in S connecting variables X and Y that contains at least one collider Z_i that is an element of Z. For the collider there are three possibilities:
   (a) $X \dots C_i \rightarrow Z_i \leftarrow C_j \dots Y$
   (b) $X \dots C_i \leftrightarrow Z_i \leftarrow C_j \dots Y$
   (c) $X \dots C_i \leftrightarrow Z_i \leftrightarrow C_j \dots Y$

Now we will show that each case implies $\neg A$:

1. Transforming S into PR only adds edges and transforms double-headed edges into single headed edges, hence the path C is still present in S and it still contains no collider. This implies that $X \perp\!\!\!\perp_{PR} Y|Z$ is false.
2. (a) The path C is still present in S together with the collider in Z_i, as it has single headed incoming edges. This implies that $X \perp\!\!\!\perp_{PR} Y|Z$ is false.
   (b) The path C is still present in S. However, the double-headed edge is transformed into a single headed edge. Depending on the topological order there are two possibilities:
   - $C_i \rightarrow Z_i \leftarrow C_j$: in this case the collider is still present in PR, this implies that $X \not\perp\!\!\!\perp_{PR} Y|Z$
   - $C_i \leftarrow Z_i \leftarrow C_j$: in this case the collider is no longer present, but in PR there is the new edge $C_i \leftarrow C_j$ and hence $X \not\perp\!\!\!\perp_{PR} Y|Z$

   (c) The path C is still present in S. However, both double-headed edges are transformed into single headed edges. Depending on the topological order there are several possibilities. For the sake of brevity we will only treat a single order here, for the others it can easily be checked that the same holds. If the order is $C_i < Z_i < C_j$, the graph becomes $C_i \rightarrow Z_i \rightarrow C_j$, but there are also edges from C_i and Z_i to C_j and its parents $Pa(C_j)$. Thus the collider is no longer present, but the extra edges ensure that $X \not\perp\!\!\!\perp_{PR} Y|Z$.

This implies that $X \perp\!\!\!\perp_{PR} Y|Z$ is false and therefore we can conclude that PR is always an I-map of S under our assumptions. □

Parametrisation. For this DAG we can use the same parametrisation as for classical BNs, i.e. learning $P(v_i|Pa(v_i))$ for each variable, where $Pa(v_i)$ denotes the parents in the new DAG. In this way the JPD over the observable variables factorises as in a classical BN, i.e. $P(v) = \prod P(v_i|Pa(v_i))$. This follows immediately from the definition of a c-component and from Equation (2).

Optimising the Parametrisation. Remark that the number of edges added during the creation of the PR-representation depends on the topological order of the SMCM.

As this order is not unique, giving precedence to variables with a lesser amount of parents, will cause less edges to be added to the DAG. This is because added edges go from parents of c-component members to c-component members that are topological descendants.

By choosing an optimal topological order, we can conserve more conditional independence relations of the SMCM and thus make the graph more sparse, leading to a more efficient parametrisation.

Note that the choice of the topological order does not influence the correctness of the representation, Theorem 1 shows that it will always be an I-map.

4.3 Probabilistic Inference

Two of the most famous existing probabilistic inference algorithms for models without latent variables are the $\lambda - \pi$ algorithm [15] for tree-structured BNs, and the *junction tree* algorithm [16] for arbitrary BNs.

These techniques cannot immediately be applied to SMCMs for two reasons. First of all until now no efficient parametrisation for this type of models was available, and secondly, it is not clear how to handle the bi-directed edges that are present in SMCMs.

We have solved this problem by first transforming the SMCM to its PR-representation which allows us to apply the junction tree (JT) inference algorithm. This is a consequence of the fact that, as previously mentioned, the PR-representation is an I-map over the observable variables. And as the JT algorithm only uses independences in the DAG, applying it to an I-map of the problem gives correct results. See Figure 3(b) for the junction tree obtained from the parametrised representation in Figure 3(a). Although this seems to be a minor improvement in this example, it has to be noted that this is the best possible results for this structure. The complexity of the junction tree in general will be dependent on the structure between the observed variables and on the complexity of the c-components.

4.4 Causal Inference

In [2], an algorithm for performing causal inference was developed, however as mentioned before they have not provided an efficient parametrisation.

In [6,7], a procedure is discussed that can identify a limited amount of causal inference queries. More precisely only those whose result is equal for all the members of a Markov equivalence class represented by a CPAG.

In [17], causal inference in AGs is shown on an example, but a detailed approach is not provided and the problem of what to do when some of the parents of a variable are latent is not solved.

By definition in the PR-representation, the parents of each variable are exactly those variables that have to be conditioned on in order to obtain the factor of that variable in the calculation of the c-component, see Table 1 and [2]. Thus, the PR-representation provides all the necessary quantitative information, while the original structure of the SMCM provides the necessary structural information, for Tian's algorithm to be applied.

5 Conclusions and Perspectives

In this paper we have introduced techniques for causal graphical modeling with latent variables. We pointed out that none of the existing techniques provide a complete answer to the problem of modeling systems with latent variables.

We have discussed concisely the structure learning process and in more detail the parametrisation of the model and probabilistic and causal inference. As the experimental structure learning approach relies on randomized controlled experiments, in general it is not scalable to problems with a large number of variables, due to the associated large number of experiments. Furthermore, it cannot be applied in application areas where such experiments are not feasible due to practical or ethical reasons.

SMCMs have not been parametrised in another way than by the entire joint probability distribution, we showed that using an alternative representation, we can parametrise SMCMs in order to perform probabilistic as well as causal inference. Furthermore this new representation allows to learn the parameters using classical methods.

We have informally pointed out that the choice of a topological order when creating the PR-representation, influences the size and thus the efficiency of the PR-representation. We would like to investigate this property in a more formal manner. Finally, we have started implementing the techniques introduced in this paper into the structure learning package (SLP)[1] of the Bayesian networks toolbox (BNT)[2] for MATLAB.

Acknowledgements

This work was partially supported by the IST Programme of the European Community, under the PASCAL network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

1. Pearl, J.: Causality: Models, Reasoning and Inference. MIT Press, Cambridge (2000)
2. Tian, J., Pearl, J.: On the identification of causal effects. Technical Report (R-290-L), UCLA C.S. Lab (2002)
3. Richardson, T., Spirtes, P.: Ancestral graph markov models. Technical Report 375, Dept. of Statistics, University of Washington (2002)
4. Eberhardt, F., Glymour, C., Scheines, R.: On the number of experiments sufficient and in the worst case necessary to identify all causal relations among n variables. In: Proc. of the 21st Conference on Uncertainty in Artificial Intelligence (UAI), pp. 178–183 (2005)

[1] http://banquiseasi.insa-rouen.fr/projects/bnt-slp/
[2] http://bnt.sourceforge.net/

5. Meganck, S., Leray, P., Manderick, B.: Learning causal bayesian networks from observations and experiments: A decision theoretic approach. In: Modeling Decisions in Artificial Intelligence. LNCS, pp. 58–69 (2006)
6. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction and Search. MIT Press, Cambridge (2000)
7. Zhang, J.: Causal Inference and Reasoning in Causally Insufficient Systems. PhD thesis, Carnegie Mellon University (2006)
8. Meganck, S., Maes, S., Leray, P., Manderick, B.: Learning semi-markovian causal models using experiments. In: Proceedings of The third European Workshop on Probabilistic Graphical Models, PGM 06 (2006)
9. Zhang, J., Spirtes, P.: A transformational characterization of markov equivalence for directed acyclic graphs with latent variables. In: Proc. of the 21st Conference on Uncertainty in Artificial Intelligence (UAI), pp. 667–674 (2005)
10. Tian, J.: Generating markov equivalent maximal ancestral graphs by single edge replacement. In: Proc. of the 21st Conference on Uncertainty in Artificial Intelligence (UAI), pp. 591–598 (2005)
11. Ali, A.R., Richardson, T., Spirtes, P., Zhang, J.: Orientation rules for constructing markov equivalence classes of maximal ancestral graphs. Technical Report 476, Dept. of Statistics, University of Washington (2005)
12. Spirtes, P., Meek, C., Richardson, T.: An algorithm for causal inference in the presence of latent variables and selection bias. In: Computation, Causation, and Discovery, pp. 211–252. AAAI Press, Menlo Park (1999)
13. Zhang, J., Spirtes, P.: A characterization of markov equivalence classes for ancestral graphical models. Technical Report 168, Dept. of Philosophy, Carnegie-Mellon University (2005)
14. Maes, S., Meganck, S., Leray, P.: An integral approach to causal inference with latent variables. In: Causality and Probability in the Sciences (Texts in Philosophy), (College Publications) (2007)
15. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco (1988)
16. Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society, series B 50, 157–244 (1988)
17. Richardson, T., Spirtes, P.: Causal inference via ancestral graph models. In: Oxford Statistical Science Series: Highly Structured Stochastic Systems, Oxford University Press, Oxford (2003)

Learning Causal Bayesian Networks from Incomplete Observational Data and Interventions

Hanen Borchani, Maher Chaouachi, and Nahla Ben Amor

LARODEC, Institut Supérieur de Gestion de Tunis, 41 Avenue de la liberté, 2000 Le Bardo, Tunisie
hanene.borchani@gmail.com, chaouachi_maher@yahoo.fr, nahla.benamor@gmx.fr

Abstract. This paper proposes a new method for learning causal Bayesian networks from incomplete observational data and interventions. We extend our Greedy Equivalence Search-Expectation Maximization (GES-EM) algorithm [2], initially proposed to learn Bayesian networks from incomplete observational data, by adding a new step allowing the discovery of correct causal relationships using interventional data. Two intervention selection approaches are proposed: an adaptive one, where interventions are done sequentially and where the impact of each intervention is considered before starting the next one, and a non-adaptive one, where the interventions are executed simultaneously. An experimental study shows the merits of the new version of the GES-EM algorithm by comparing the two selection approaches.

1 Introduction

Bayesian networks are becoming a popular tool for representing uncertainty in artificial intelligence [12,17]. They have been implemented in several real applications in different areas such as medical diagnosis, pattern recognition, credit assessment and fraud detection. Recently, a lot of research has been oriented towards learning those models from data. There are two main approaches: The *score-based* approach [3,5,6,10] which attempts to identify the network that maximizes a scoring function evaluating how well the network fits the data, and the *constraint-based* one [18,19], which performs statistical tests to determine conditional independence relationships among variables in the given data, and then searches for a network consistent with these assumed relationships.

Since they perform learning from purely *observational data*, both approaches suffer from a substantial problem consisting in the fact that they cannot distinguish between equivalent networks, and thus, cannot identify correctly the causal network. In fact, learning about causal relationships is crucial for at least two reasons, namely, it allows us to gain more understanding about the problem domain, and to make predictions in the presence of interventions [19].

To learn causal networks, *interventional data*, i.e. samples conditioned on the particular values of one or more variables that have been experimentally manipulated, are required. Given such data, interventions allowing us to discover

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 17–29, 2007.

the correct causal relationships should be carefully selected. In other words, one should decide which interventions are deemed most informative while respecting available resources.

This paper proposes a new approach for learning causal Bayesian networks from incomplete observational data, where some values of some variables are missing, as well as interventional data. Indeed, the observational data may be incomplete for several reasons such as noisy measurements, equipment malfunction or not entered values due to misunderstanding.

Roughly speaking, as first step, our approach makes use of the GES-EM algorithm [2] to identify the Bayesian network equivalence class that best matches the incomplete observational data. Next, in order to infer the causal Bayesian network, we propose to extend this algorithm by adding a new step that allows us to select efficiently appropriate interventions, and therefore, to discover properly the causal relationships.

Two intervention selection approaches will be studied, namely an *adaptive* one, where interventions are done sequentially, so the impact of each intervention is considered before starting the next one, and a *non-adaptive* approach, where the whole selected interventions are executed simultaneously.

The remainder of this paper is organized as follows: Section 2 provides some notations and recalls basics of Bayesian networks as well as causal Bayesian networks. Section 3 describes the GES-EM algorithm used for learning Bayesian network equivalence classes from incomplete observational data. Section 4 proposes our new criteria and approaches to select efficiently appropriate interventions and infer a correct causal Bayesian network. Finally, Section 5 describes experimental study showing the merits of the new version of GES-EM and comparing the two proposed selection approach.

2 Background and Notations

The following syntactical conventions are used: $\mathbf{U} = \{X_1, ..., X_n\}$ denotes the universe defined as a finite set of n discrete random variables. A variable is denoted by an upper case letter (e.g. X, Y, X_i) and a state or value of that variable by the same lower-case letter (e.g. x, y, x_i). A set of variables is denoted by a bold-face capitalized letter (e.g. $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Pa}(X_i)$) and the corresponding bold-face lower-case letter (e.g. $\mathbf{x}$, $\mathbf{y}$, $\mathbf{pa}(X_i)$) denotes an assignment or state for each variable in a given set. Calligraphic letters (e.g. $\mathcal{B}$, $\mathcal{CB}$, $\mathcal{G}$, $\mathcal{E}$) denotes statistical models both parameterized and non-parameterized.

2.1 Bayesian Networks

A Bayesian network $\mathcal{B}$ [12,17] is a pair ($\mathcal{G}$, $\mathbf{\Theta}$). The first component is the structure $\mathcal{G}$ which is a Directed Acyclic Graph (DAG), where the set of vertices or nodes represents the variables, and the set of directed edges or arcs corresponds to dependence relationships between these variables. The second component, namely the set of parameters $\mathbf{\Theta} = \{\mathbf{\Theta}_1, \mathbf{\Theta}_2, ..., \mathbf{\Theta}_n\}$, specifies all the conditional

probability distributions. $\mathbf{\Theta}_i = P(X_i \mid \mathbf{Pa}(X_i))$ denotes the conditional probability distribution of each node X_i given its parents, denoted by $\mathbf{Pa}(X_i)$.

Two Bayesian network structures $\mathcal{G}$ and $\mathcal{G}'$ are said to be equivalent, denoted by $\mathcal{G} \approx \mathcal{G}'$, if they can be used to represent the same set of probability distributions. More formally, Chickering [4] provides the following definition:

Definition 1. *Two DAGs $\mathcal{G}$ and $\mathcal{G}'$ are equivalent if for every Bayesian network $\mathcal{B} = (\mathcal{G}, \mathbf{\Theta})$, there exists a Bayesian network $\mathcal{B}' = (\mathcal{G}', \mathbf{\Theta}')$ such that $\mathcal{B}$ and $\mathcal{B}'$ define the same probability distribution, and vice versa.*

Moreover, Verma and Pearl [21] propose the following definition which determines graphically the equivalence of two *DAGs*.

Definition 2. *Two DAGs are equivalent if and only if they have the same skeletons (i.e. the undirected graph resulting from ignoring the directionality of every edge) and the same v-structures (i.e. ordered triples of nodes (X, Y, Z), such that $X \rightarrow Y$, $Z \rightarrow Y$ and X and Z are not adjacent).*

Each set of equivalent *DAGs* defines an *equivalence class of Bayesian networks*, denoted by $\mathcal{E}$, which can be represented via a unique *Completed Partially Directed Graph* (*CPDAG*), denoted by $\mathcal{P}^c$ [5]. A Partially Directed Acyclic Graph (*PDAG*) is a graph that contains both directed and undirected edges. A *compelled* edge is an edge that exists with the same orientation for every *DAG* member of an equivalence class. If an edge is not compelled, then it is *reversible*. Thus, a *CPDAG* $\mathcal{P}^c$ corresponding to an equivalence class $\mathcal{E}$ is a *PDAG* consisting of a directed edge for every compelled edge in the equivalence class, and an undirected edge for every reversible edge in the equivalence class [5].

2.2 Causal Bayesian Networks

Causal Bayesian networks, denoted by $\mathcal{CB}$, are Bayesian networks with some particular properties concerning their interpretation [18,19]. More precisely, the parent set of a given variable is seen as its immediate cause, and consequently, equivalence hypothesis of Bayesian networks (expressed by definitions 1 and 2) is not valid any more. For instance, although the two Bayesian networks $X \rightarrow Y$ and $X \leftarrow Y$ are equivalent, only one of them is a correct causal Bayesian network. In fact, if we consider the first network X causes Y, then, manipulating the value of X affects the value of Y. However, if we consider the second one, Y is a cause of X, then manipulating X will not affect Y. To more illustrate this purpose we can consider the famous link Asia−Tuberculosis in the Asia network [13]. It is clear that visiting Asia may cause Tuberculosis, while having a Tuberculosis has no impact on visiting Asia. Thus, a causal network is a proper Bayesian network but the contrary is not always true. This means that its structure is more comprehensible and more expressive than the standard Bayesian network. Furthermore, given a causal network one can use it in different manners depending on the requirement. In fact, if we want to determine how the observation of specific values (evidence) affects the probabilities of query variable(s), then we can use *probabilistic inference* [12,17], while if the objective is to predict the effect of an intervention on

the remaining variables, then we should use *causal inference* [14]. Obviously, if we apply causal inference on standard Bayesian networks, then results can be biased.

3 Learning Bayesian Network Equivalence Classes from Incomplete Observational Data: GES-EM Algorithm

Learning Bayesian networks from observational data is an unsupervised learning problem, which aims to determine a network, or set of networks, that best fit this data. Most learning methods, to date, are based on the assumption that the observational data are complete, that is, the values of all variables are observed for each record in the data [3,5,6,10]. Unfortunately, this assumption is unrealistic since most real-world data rarely conform to the ideal of being complete, and quite often involve missing information.

In [2], we have proposed a novel approach, named Greedy Equivalence Search-Expectation Maximization (GES-EM), for learning Bayesian network equivalence classes from incomplete data, and proved theoretically and experimentally its efficiency. In fact, the main idea of this approach, is to alternate between Expectation Maximization (EM) [7] iterations, which optimize the parameters for the current equivalence class, and structure search iterations, which aim to identify the possible neighbors of the current equivalence class using the Greedy Equivalence Search (GES) algorithm [5].

Roughly speaking, GES-EM algorithm starts with an initial equivalence class $\mathcal{E}^0$, containing a unique empty Bayesian network represented via the structure $\mathcal{G}^0$, and the randomly initialized parameter set $\mathbf{\Theta}^0$, then, it executes a single edge insertion phase followed by a single edge deletion one.

These two phases are quite analogous, thus, we only present the single edge insertion phase outlined by the following algorithm:

Algorithm 1. *Single edge insertion phase*
begin
  $i \leftarrow 0$; *Convergence* $\leftarrow$ *false*; *Stop* $\leftarrow$ *false*;
  while (*not Convergence*) *and* (*not Stop*) **do**
    1. Run EM, using $\mathcal{G}^i$ *and* $\mathbf{\Theta}^i$, *producing* $\mathbf{\Theta'^i}$;
    2. Generate $\mathbf{P}^+(\mathcal{E}^i)$ *via applying all valid Insert operators to* $\mathcal{E}^i$;
    3. $\mathbf{G}^+(\mathcal{E}^i) \leftarrow$ ***PDAG-to-DAG***$(\mathbf{P}^+(\mathcal{E}^i))$;
    4. **if** $\mathbf{G}^+(\mathcal{E}^i) \neq \emptyset$ **then**
      4.1. Compute $Q(\mathcal{G}, \mathbf{\Theta} : \mathcal{G}^i, \mathbf{\Theta'^i})$ *for each* $\mathcal{G} \in \mathbf{G}^+(\mathcal{E}^i)$;
      4.2. $(\mathcal{G}^{i+1}, \mathbf{\Theta}^{i+1}) \leftarrow argmax\ Q(\mathcal{G}, \mathbf{\Theta} : \mathcal{G}^i, \mathbf{\Theta'^i})$;
      4.3 **if** $Q(\mathcal{G}^{i+1}, \mathbf{\Theta}^{i+1} : \mathcal{G}^i, \mathbf{\Theta'^i}) > Q(\mathcal{G}^i, \mathbf{\Theta'^i} : \mathcal{G}^i, \mathbf{\Theta'^i})$ **then**
        $\mathcal{E}^{i+1} \leftarrow$ ***DAG-to-CPDAG***$(\mathcal{G}^{i+1})$;
        $i \leftarrow i + 1$;
      4.4 **else**
        Convergence $\leftarrow$ *true*;
    5. **else**
      Stop $\leftarrow$ *true*;
end

Each iteration in this phase starts by running the EM algorithm [7] to convergence, in order to find improved parameter values $\boldsymbol{\Theta}'^{i}$ for the current structure $\mathcal{G}^i$ (step 1). Next, it applies all valid *Insert* operators [5] to $\mathcal{E}^i$ in order to obtain the set of its neighbors (step 2). This set is denoted by $\mathbf{P}^+(\mathcal{E}^i)$ since each resulting neighbor is a *PDAG*, denoted $\mathcal{P}$, not necessarily completed.

Then, each neighbor $\mathcal{P} \in \mathbf{P}^+(\mathcal{E}^i)$ is converted into its *consistent extension* $\mathcal{G}$ defined as a *DAG* having the same skeleton and the same set of v-structures as $\mathcal{P}$, and also the same orientation of every directed edge in $\mathcal{P}$. This conversion step is accomplished via the *PDAG-to-DAG* algorithm [8] (step 3).

The set of all obtained consistent extensions is denoted by $\mathbf{G}^+(\mathcal{E}^i)$. If this set is empty, the first phase ends and the second one starts from the current state (step 5). Otherwise, each neighbor $\mathcal{G} \in \mathbf{G}^+(\mathcal{E}^i)$ is evaluated using the expected Bayesian Information Criterion (BIC) score, denoted Q [9] (step 4.1).

Next, the highest scoring neighbor is selected (step 4.2) and compared to the one found in the previous iteration. If it is lower, then the first phase ends and the second one proceeds from the previous best state (step 4.4). Otherwise, the best equivalence class is generated, via converting the best selected consistent extension $\mathcal{G}$ to a *CPDAG*, using the conversion algorithm *DAG-to-CPDAG* [5] (step 4.3). The obtained equivalence class becomes the current state and the search continues until no neighbors are generated or a local maximum is reached.

The second phase of GES-EM, i.e. the single edge deletion phase, is executed in the same manner as the single edge insertion phase by replacing *Insert* operators, in step 2, by *Delete* ones [5].

When the GES-EM ends, it returns the equivalence class which better fits the initial data. Then, according to the equivalence definition, we can choose one of the equivalent networks pertaining to it. This choice is generally done in an arbitrary manner as proposed by [8]. Nevertheless, from a causal point of view, this can compromise the network accuracy since we are not sure to return the correct causal network. In fact, we should properly orient existing edges in order to express causal influences between variables.

Obviously, this cannot be accomplished using the already explored observational data, and requires some additional information, known as interventional data. Hence, using such data, the main idea is to extend the GES-EM algorithm and correctly orient reversible edges in the returned equivalence class via a real intervention selection strategy.

4 Selecting Appropriate Interventions

By interventions, we mean experiment, action or manipulation performed on a given variable by fixing it to a single value and observing its impact on target variables. The main issue here is *how to select appropriate interventions?*

To respond to this question, we first define intervention selection criteria based on connectivity and costs, then, we propose two selection approaches

optimizing these criteria: In the first one, which is *adaptive*, interventions are executed sequentially by considering at each step the result of previous interventions. While, in the second approach, which is *non-adaptive*, interventions are done simultaneously.

Note that, within both approaches, each intervention execution is followed by applying the PC rules [19] in order to infer directions of some additional edges, which are not directly connected to the manipulated node. The PC rules can be summarized as follows:

- **R1**: *Directing edges without introducing new v-structures*:
  $\forall X_i, X_j \in U$, if $X_i \rightarrow X_j$ and X_j and X_k are adjacent, X_i and X_k are not, and there is no arrow into X_j then orient X_j-X_k as $X_j \rightarrow X_k$
- **R2**: *Directing edges without introducing cycles*:
  $\forall X_i, X_j \in U$, if it exists a directed path between X_i and X_j, and an edge $X_i - X_j$, then orient it as $X_i \rightarrow X_j$.

4.1 Selection Criteria

It is clear that the selection of appropriate interventions will be made within nodes having *non-oriented neighbors* (i.e neighbors which are neither parents nor children). Thus, we will split the node set $\mathbf{U}$ into two subsets $\mathbf{U}^+$ (nodes with no edges) and $\mathbf{U}^-$ (nodes with at least one edge) and only consider nodes in $\mathbf{U}^-$. For each of these nodes, we will define two criteria, the first is a topological one relative to its connectivity, and the second concerns the practical aspect of the intervention and more precisely to its cost.

Connectivity. Since our objective is to fully orient the $CPDAG$ while minimizing interventions, the first considered criterion is the connectivity of each node in $\mathbf{U}^-$. In fact, it seems obvious to manipulate variables susceptible to orient the maximum number of edges, and hence to consider the connectivity as a suitable criterion to insure this task. When manipulating a node X_i, we insure certainly the orientation of its connected edges. However, it is difficult to predict the impact of the intervention on the rest of the graph. In fact, an intervention on a node with a small number of connected edges can orient the whole graph as well as (and even better than) a one relative to a node having a great number of connected edges.

Example 1. *Let us consider the DAG in the Figure 1(a). Suppose that the GES-EM algorithm generates the CPDAG of Figure 1(b). It is clear that the node X_3 has the highest number of connected edges, nevertheless, its manipulation only orient the edges X_3-X_1, X_3-X_4, X_3-X_7, and X_3-X_8 (after applying PC rules). However, despite its lower number of connected edges, an intervention on X_1 will orient the whole graph.*

Meganck et al. [15] have proposed to use the connectivity criterion by only considering the number of connected edges relative to each node $X_i \in \mathbf{U}^-$. Our

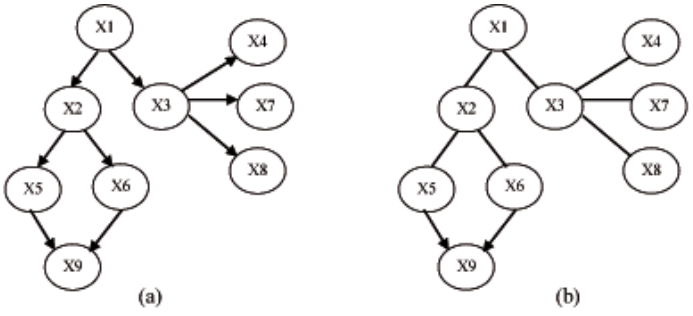


Fig. 1. (a) Example of *DAG* (b) Example of resulted CPDAG from GES-EM algorithm

idea is to also consider edges connected to neighbors. More precisely, for each node $X_i \in \mathbf{U}^-$ we will consider its connected edges (this first level is denoted by S_{1i}), and also those relative to its neighbors except those linking them to it (this second level is denoted by S_{2i}). This choice is motivated by the fact that each node is conditionally independent of all other nodes given its *Markov blanket*, i.e. the set consisting of its parents, its children, and the other parents of its children [17]. Hence, the connectivity criterion for each node $X_i \in \mathbf{U}^-$ can be expressed by the following weight:

$$W(X_i) = \mid S_{1i} \mid + \alpha \mid S_{2i} \mid \tag{1}$$

where $\alpha \in [0, 1[$ is a calibrating coefficient relative to the importance that we give to edges pertaining to the second level. The greater α is, the more the second level is considered. Note that, the value 1 is excluded since the first level should be always favored. Moreover, if $\alpha = 0$ then we recover Meganck et al. criterion [15]. The value of α can be fixed by an expert regarding the $CPDAG$ topology.

Example 2. *Let us reconsider the CPDAG presented by Figure 1(b) where* $\mathbf{U}^+ = \{X_9\}$ *and* $\mathbf{U}^- = \{X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8\}$. *Table 1 gives the connectivity weights with* $\alpha = 0.6$.

Table 1. Connectivity weights relative to the CPDAG of Figure 1(b)

Node	S_1	S_2	W
X_1	X_1-X_2, X_1-X_3	X_3-X_4, X_3-X_7, X_3-X_8, X_2-X_5, X_2-X_6	5.0
X_2	X_2-X_1, X_2-X_5, X_2-X_6	X_1-X_3	3.6
X_3	X_3-X_1, X_3-X_4, X_3-X_7, X_3-X_8	X_1-X_2	4.6
X_4	X_4-X_3	X_3-X_1, X_3-X_7, X_3-X_8	2.8
X_5	X_5-X_2	X_2-X_1, X_2-X_6	2.2
X_6	X_6-X_2	X_2-X_1, X_2-X_5	2.1
X_7	X_7-X_3	X_3-X_1, X_3-X_4, X_3-X_8	2.8
X_8	X_8-X_3	X_3-X_1, X_3-X_4, X_3-X_7	2.8

Note that X_1 has the higher connectivity weight, and thus, it will be interesting to choose it. In fact, regarding the real graph in Figure 1(a) the manipulation of X_1 will orient the whole graph.

Intervention cost. In real application, interventions do not always have the same cost, moreover some of them, such as provoking earthquake or making somebody sick, are unrealistic. Thus, it is interesting to consider the intervention cost as a supplementary selection criterion by assigning to each node $X_i \in \mathbf{U}^-$ a cost, denoted by $Cost(X_i)$ such that impossible interventions are marked by an infinite cost (i.e. ∞). For the sake of simplicity, we consider that $Cost(X_i)$ is relative to both experimentation and evaluation costs on X_i.

4.2 Optimal Selection of Interventions

Regarding the connectivity and the intervention cost criteria, our objective is to select best interventions, i.e. those susceptible to orient the maximal number of edges while minimizing the cost. To this end, we propose two different approaches: the first is *adaptive*, where interventions are done sequentially, and the second is *non-adaptive*, where the whole interventions are executed simultaneously. The selection phase (even adaptive or not) should be added as an additional step to the GES-EM algorithm after the single edge insertion and deletion phases in order to enable it to correctly learn causal relationships.

Adaptive Selection Approach. In general, adaptive approaches [15,16,20] search the most informative intervention to perform, following the impact of the previously executed one. In this approach, we show as well how to make use of the already defined selection criteria in order to properly choose best interventions. In fact, we take as input the $CPDAG$ resulting from the GES-EM algorithm, the list of node costs pertaining to $\mathbf{U}^-$ (denoted by Cost), and the total budget that the user is able to invest (denoted by Budget). Then, we select best intervention to perform according to the cost and connectivity criteria. The key idea consists in calibrating the weight of each node $X_i \in \mathbf{U}^-$ by its manipulation cost as follows:

$$V(X_i) = \frac{W(X_i)}{Cost(X_i)} \tag{2}$$

The intervention should be done on the node maximizing (2). The PC rules are, then, applied to infer directions of some additional edges. The algorithm stops when the whole graph is oriented, or if it is impossible to perform more interventions due to the budget limitation. In this case, an expert can be consulted to orient the remaining edges, or adjust the assigned budget. The outline of the adaptive algorithm is as follows:

Algorithm 2. *Adaptive algorithm*
***Input**: CPDAG $\mathcal{P}^c$, Cost, Budget - **Output**: Causal Bayesian network $\mathcal{CB}$*
begin
  repeat
    $\forall X_i \in \mathbf{U}^-$, *compute* $V(X_i)$;
    Perform experiment on X_B *such that* $V(X_B) = max_{X_i \in \mathbf{U}^-}(V(X_i))$;
    Budget = Budget - Cost(X_B);
    for $all X_i \in S_{1B}$ **do**
      if X_i changes **then** *Orient* $X_B - X_i$ *as* $X_B \rightarrow X_i$;
      else *Orient it as* $X_B \leftarrow X_i$;
  until $\mathbf{U}^- = \emptyset$ or $\forall X_i \in \mathbf{U}^-$, $Budget < Cost(X_i)$;
  return $\mathcal{CB}$;
end

Meganck et al. [15] have addressed the selection problem by proposing several adaptive optimization approaches based on the evaluation function expressed by (2) with $\alpha = 0$ (i.e. by just considering direct edges). We can mention, in particular, the optimistic and the pessimistic ones, namely *MaxiMax* and *MaxiMin* approaches. Their principle is to first study different scenarios regarding each intervention, then to select, with *MaxiMax*, the one that may direct the most edges and with *MaxiMin*, the one that ensures the minimal number of inferred edges.

The experimental study, given in Section 5, shows that our approach is competitive with these approaches with lower theoretical complexity.

Non-Adaptive Selection Approach. In this approach, we adapt our selection criteria to choose and perform simultaneously the best set of interventions. Therefore, the fundamental problem is how to select carefully this set in order to orient efficiently the graph while respecting the available budget.

Such a problem can be formulated as a *knapsack problem*, where our objective is to maximize the sum of node weights relative to the set of interventions, while taking into account both the budget and the neighboring constraints. In fact, the intervention set should not contain adjacent nodes in order to avoid useless interventions.

More formally, this optimization problem can be formulated as follows:

$$Max \sum_{i}^{n} C_i * W(X_i) \tag{3}$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^{n} C_i * Cost(X_i) \leq Budget \\ \forall X_i, X_j \in \mathbf{U}^- \text{ s.t. } X_i \text{ is a neighbor of } of X_j, C_i + C_j \leq 1 \\ C_i \in \{0, 1\}. \end{cases}$$

where $C_i = 1$ (resp. $C_i = 0$) means that X_i is selected (resp. is not selected) in the intervention set, denoted by $\mathbf{U}^{Kn}$. The non adaptive algorithm can be outlined as follows:

Algorithm 3. *Non-adaptive algorithm*
***Input**: CPDAG $\mathcal{P}^c$, Cost, Budget -* ***Output**: Causal Bayesian network $\mathcal{CB}$*
begin
Construct the interventional set $\mathbf{U}^{Kn}$ *using (3);*
Perform simultaneous experiments $\forall X_i \in \mathbf{U}^{Kn}$*;*
for $allX_i \in \mathbf{U}^{Kn}$ **do**
for $allX_j \in S_{1i}$ **do**
if X_j changes **then** *Orient* $X_i - X_j$ *as* $X_i \rightarrow X_j$*;*
else *Orient it as* $X_i \leftarrow X_j$*;*
Apply PC rules R1 and R2 until no more edges can be oriented;
return $\mathcal{CB}$*;*
end

5 Experimental Study

In this section, we present a comparative study between adaptive an non-adaptive approaches. We also compare our adaptive heuristic with those proposed by Meganck et al. [15] i.e. *MaxiMax* and *MaxiMin*. To this end, we considered the well-known Insurance network [1] with 27 nodes and 52 arcs. The experimental data were generated as follows:

- *Observational data*: we randomly sampled observational data sets of different sizes (i.e. #ODS= 500, 1000, 5000) using the probabilistic logic sampling method [11] on the Insurance network. Then, we randomly removed values from each data set in order to get incomplete observational data sets with various missing value percent (i.e. %mv).
- *Interventional data*: since real interventional data are not available, we generate them in a synthetic manner, by mutilating the original network. More precisely, each intervention on a selected node is simulated by cutting all its incoming arcs and then by applying the **do**() operator [18]. We approximately generate 1000 causal instances.
- *Intervention costs*: we have affected different costs in a logical manner by assigning infinite costs to impossible interventions, such as *age manipulation*.

The comparison is based on three criteria: The total number of the performed interventions, their cost and the percentage of oriented edges. Table 2 summarizes obtained experimental results.

It is clear that both adaptive and non-adaptive approaches improve the GES-EM algorithm since they allow a correct orientation of edges instead of a random one.

A deep analysis of results relative to adaptive approaches (i.e. *Weight*, *MaxiMax*, *MaxiMin*) shows that they are competitive with respect to the three evaluation criteria. In fact, they consume almost the same budget with close interventions number and they provide the same improvements (i.e. percentage of oriented edges).

Table 2. Experimental results over Insurance network (values given in brackets are relative to a limited budget, i.e. Budget= 120, remaining values are relative to an unlimited budget, i.e. Budget= ∞)

	Adaptive learning			**Non adaptive learning**
	Weight	**MaxiMax**	**MaxiMin**	
#ODS=500, %mv=30, # initial edge =12				
Total cost	140(90)	130(110)	130(80)	250(120)
Intervention number	4(3)	4(3)	3(2)	6(3)
% oriented edge	100(91.6)	100(91.6)	100(91.6)	100(66.6)
#ODS=1000, %mv=20, # initial edge =16				
Total cost	120(120)	140(110)	120(120)	220(120)
Intervention number	4(4)	5(4)	4(4)	6(3)
% oriented edge	100(100)	100(93.7)	100(100)	100(75.0)
#ODS=5000, %mv=20, # initial edge =15				
Total cost	1230(80)	1210(110)	1230(80)	220(100)
Intervention number	5(2)	5(3)	5(2)	6(3)
% oriented edge	93.3(66.6)	86.6(80.0)	86.6(66.6)	86.6(73.3)

Nevertheless, the *MaxiMax* and the *MaxiMin* approaches present a high theoretical complexity (i.e. $O(n * 2^k)$ where n is the number of nodes in the $CPDAG$ and k is the maximum number of edges relative to any $X_i \in \mathbf{U}$) contrary to our approach which is linear (i.e. O(n)) since it only computes the weight of each node without considering different configurations of remaining links.

Experiments also show an additional interesting result regarding adaptive and non adaptive approaches. In fact, it is clear that adaptive ones present better results. This is unsurprising since the vocation of non-adaptive approaches is to satisfy the objective function (i.e. maximizing connectivity weights) while respecting the initial budget without considering eventual interactions between interventions. Such a strategy can be adopted when the user has a fixed budget with a possibility of multi-agent interventions since it allows him to save execution time.

6 Conclusion

This paper proposes a novel approach for learning causal Bayesian networks from incomplete observational data and interventions. The basic idea of our approach is to extend the GES-EM [2] algorithm via performing an additional phase in order to discover causal relationships. This phase is crucial since it permits us to properly orient the reversible edges in the learned $CPDAG$.

More precisely, we have proposed two intervention selection approaches: an adaptive one, where interventions are performed sequentially and where the impact of each intervention is considered before starting the next one, and a non-adaptive approach, where interventions are executed simultaneously.

Obviously, the quality of the $CPDAG$ issued from the GES-EM algorithm have a direct impact on the final result. Actually we assume that the GES-EM algorithm provides a correct $CPDAG$ with respect to observational data. This is not always true, in fact, if the initial $CPDAG$ contains an edge $X_i - X_j$, such that an intervention on X_i has no effect on X_j and vice versa, this means that this link is wrong. In the current work, we save such edges, but an interesting future work will be to revise the $CPDAG$ using both observational and experimental data.

Another line of research would be to consider incomplete interventional data as it is the case in observational data. In fact, in some situations an experiment can give us some missing values, thus it will be interesting to consider them instead of rejecting them.

References

1. Binder, J., Koller, D., Russell, S., Kanazawa, K.: Adaptive Probabilistic Networks with Hidden Variables. Machine Learning 29, 213–244 (1997)
2. Borchani, H., Ben Amor, N., Mellouli, K.: Learning Bayesian Network Equivalence Classes from Incomplete data. In: Todorovski, L., Lavrač, N., Jantke, K.P. (eds.) DS 2006. LNCS (LNAI), vol. 4265, pp. 291–295. Springer, Heidelberg (2006)
3. Buntine, W.L.: A Guide to the Literature on Learning Probabilistic Networks from Data. IEEE Transactions on Knowledge and Data Engineering 8, 195–210 (1996)
4. Chickering, D.M.: A Transformational Characterization of Equivalent Bayesian Networks. In: the Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 87–98 (1995)
5. Chickering, D.M.: Optimal Structure Identification With Greedy Search. Journal of Machine Learning Research 3, 507–554 (2002)
6. Cooper, G.F., Herskovits, E., Bayesian, A.: Method for the Induction of Probabilistic Networks from Data. Machine Learning 9, 309–347 (1992)
7. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM algorithm. Journal of the Royal Statistical Society B39, 1–38 (1977)
8. Dor, D., Tarsi, M.: A Simple Algorithm to Construct a Consistent Extension of a Partially Oriented Graph. Technical Report R-185, Cognitive Systems Laboratory, UCLA Computer Science Department (1992)
9. Friedman, N.: Learning Belief Networks in the Presence of Missing Values and Hidden Variables. In: Fourteenth Conference of Machine Learning, pp. 125–133 (1997)
10. Heckerman, D.: A Tutorial on Learning Bayesian Networks. Technical Report MSR-TR-95-06, Microsoft Research (1995)
11. Henrion, M.: Propagating uncertainty in Bayesian networks by probabilistic logic sampling. Uncertainty in Artificial Intelligence 2, 149–163 (1988)
12. Jensen, F.V.: An Introduction to Bayesian Networks. UCL Press, London (1996)
13. Lauritzen, S.L., Spiegelhalter, D.J.: Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems. Journal of the Royal Statistical Society 50(2), 157–224 (1988)
14. Lauritzen, S.L.: Causal Inference from Graphical Models. In: Complex Stochastic Systems, pp. 63–107 (2001)

15. Meganck, S., Leray, P., Manderick, B.: Learning Causal Bayesian Networks from Observations and Experiments: A Decision Theoritic Approach. In: Torra, V., Narukawa, Y., Valls, A., Domingo-Ferrer, J. (eds.) MDAI 2006. LNCS (LNAI), vol. 3885, pp. 58–69. Springer, Heidelberg (2006)
16. Murphy, K.P.: Active Learning of Causal Bayes Net Structure. Technical report, Department of Computer Science, UC Berkeley (2001)
17. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)
18. Pearl, J.: Causality: Models, Reasoning and Inference. Cambridge University Press, Cambridge (2000)
19. Spirtes, B., Glymour, C., Scheines, R.: Causation, Prediction and Search. Springer, New York (1993)
20. Tong, S., Koller, D.: Active Learning for Structure in Bayesian Networks. In: Seventeenth International Joint Conference on Artificial Intelligence, pp. 863–869 (2001)
21. Verma, T., Pearl, J.: Equivalence and Synthesis of Causal Models. In: Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence (1990)

Measuring Inconsistency for Description Logics Based on Paraconsistent Semantics *

Yue Ma[1,2], Guilin Qi[2], Pascal Hitzler[2], and Zuoquan Lin[1]

[1]Department of Information Science, Peking University, China
[2]AIFB, Universität Karlsruhe, Germany
{mayue,lz}@is.pku.edu.cn, {yum,gqi,hitzler}@aifb.uni-karlsruhe.de

Abstract. In this paper, we present an approach for measuring inconsistency in a knowledge base. We first define the degree of inconsistency using a four-valued semantics for the description logic $\mathcal{ALC}$. Then an ordering over knowledge bases is given by considering their inconsistency degrees. Our measure of inconsistency can provide important information for inconsistency handling.

1 Introduction

Inconsistency has often been viewed as erroneous information in a knowledge base, but this is not necessarily the best perspective on the problem. The study of inconsistency handling in Artificial Intelligence indeed has a long tradition, and corresponding results are recently being transferred to description logics which are a family of decidable subsets of first-order logic.

There are mainly two classes of approaches to dealing with inconsistent description logic based knowledge bases. The first class of approaches is to circumvent the inconsistency problem by applying a non-standard reasoning method to obtain meaningful answers [1,2] – i.e. to ignore the inconsistency in this manner. The second class of approaches to deal with logical contradictions is to resolve logical modeling errors whenever a logical problem is encountered [3,4].

However, given an inconsistent knowledge base, it is not always clear which approach should be taken to deal with the inconsistency. Another problem is that when resolving inconsistency, there are often several alternative solutions and it would be helpful to have some extra information (such as an ordering on elements of the knowledge base) to decide which solution is the best one. It has been shown that analyzing inconsistency is helpful to decide how to act on inconsistency [5], i.e. whether to ignore it or to resolve it. Furthermore, measuring inconsistency in a *knowledge base* in classical logic can provide some context information which can be used to resolve inconsistency [6,7,8].

* We acknowledge support by the China Scholarship Council(http://www.csc.edu.cn/), by the German Federal Ministry of Education and Research (BMBF) under the SmartWeb project (grant 01 IMD01 B), by the EU under the IST project NeOn (IST-2006-027595, http://www.neon-project.org/), and by the Deutsche Forschungsgemeinschaft (DFG) in the ReaSem project.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 30–41, 2007.

There are mainly two classes of inconsistency measures in classical logic. The first class of measures is defined by the number of formulas which are responsible for an inconsistency, i.e. a knowledge base in propositional logic is more inconsistent if more logical formulas are required to produce the inconsistency [9]. The second class considers the propositions in the language which are affected by the inconsistency. In this case, a knowledge base in propositional logic is more inconsistent if more propositional variables are affected by the inconsistency [6,10]. The approaches belonging to the second class are often based on some paraconsistent semantics because we can still find models for inconsistent knowledge bases using paraconsistent logics.

Most of the work on measuring inconsistency is concerned with knowledge bases in propositional logic. In [11], the authors generalized the work on measuring inconsistency in quasi-classical logic to the first-order case with restriction to prenex conjunctive form (PCNF) since all first-order theories can be translated into PCNF. However, it is still not clear how to properly perform PCNF on description logics (DLs) while maintaining DLs structures.

The main contributions of this paper are summarized as follows:

- We present an approach for measuring inconsistency of a DL knowledge base.
- We define domain-dependent inconsistency for a consistent knowledge base. This makes it possible to measure the inconsistency degree of a consistent DL knowledge bases with respect to a domain.
- An ordering is given which provides a way to order all knowledge bases according to their inconsistency degree. With respect to such an ordering, consistent knowledge bases are always less inconsistent than all inconsistent knowledge bases.

At the same time, there are potential applications for inconsistency measures for knowledge bases, as they provide evidences for reliability of knowledge bases when an inconsistency occurs. In a scenario where knowledge bases are merged together, we can give higher priority to knowledge bases which are less inconsistent. When resolving inconsistency in the merged knowledge base, we can delete or weaken some axioms from the knowledge base with lower priority.

In this paper, we propose an approach for measuring inconsistency in description logic based knowledge bases. We first define the degree of inconsistency using a four-valued semantics for description logic $\mathcal{ALC}$. By analyzing the degree of inconsistency of a knowledge base, we can either resolve inconsistency if the degree is high (e.g. greater than 0.7) or ignore it otherwise. After that, an ordering over inconsistent knowledge bases is given by considering their inconsistency degrees.

This paper is organized as follows. We first provide some basic notions for Description Logics in Section 2. Then, the concept of domain-dependent (in)consistency is defined in Section 3. Our measure of inconsistency is then given in Section 4. Finally, we discuss related work and conclude the paper in Section 5.

2 Preliminaries

2.1 The Description Logic $\mathcal{ALC}$

We briefly review the terminology of the description logic $\mathcal{ALC}$ and its relation with first order logic FOL. For comprehensive background reading, please refer to [12,13].

Table 1. Syntax and semantics of $\mathcal{ALC}$ and translation from $\mathcal{ALC}$ to FOL

Constructor	**Syntax**	$\Phi(C,x)$	**Semantics**
atomic concept A	A	$A(x)$	$A^I \subseteq \Delta^I$
abstract role R_A	R	$R(x,y)$	$R^I \subseteq \Delta^I \times \Delta^I$
individuals I	o	o	$o^I \in \Delta^I$
conjunction	$C_1 \sqcap C_2$	$\Phi(C_1,x) \cap \Phi(C_2,x)$	$C^I \cap D^I$
disjunction	$C_1 \sqcup C_2$	$\Phi(C_1,x) \cup \Phi(C_2,x)$	$C^I \cup D^I$
negation	$\neg C$	$\neg A(x)$	$\Delta^I \setminus C^I$
exists restriction	$\exists R.C$	$\exists y R(x,y) \wedge \Phi(C,y)$	$\{x \mid \exists y, (x,y) \in R^I \text{ and } y \in C^I\}$
value restriction	$\forall R.C$	$\forall y R(x,y) \rightarrow \Phi(C,y)$	$\{x \mid \forall y, (x,y) \in R^I \text{ implies } y \in C^I\}$

Axiom Name	**Syntax**	$\Phi(\cdot)$	**Semantics**
concept inclusion	$C_1 \sqsubseteq C_2$	$\forall x, \Phi(C_1,x) \rightarrow \Phi(C_2,x)$	$C_1^I \subseteq C_2^I$
concept assertion	$C(a)$	$\Phi(C,a)$	$a^I \in C^I$
role assertion	$R(a,b)$	R(a,b)	$(a^I, b^I) \in R^I$

Corresponding to monadic predicates, dyadic predicates, and functional constants, **concept, role,** and **individual** are fundamental notions of description logics. We assume that we are given a set of concept names (i.e., atomic unitary predicates), a set of role names (i.e., atomic binary predicates) and a set of individuals (i.e., functional constants). Complex concepts (complex monadic formulae) in $\mathcal{ALC}$ can be formed from these inductively as follows.

1. All atomic concept are concepts;
2. If C, D are concepts, then $C \sqcup D$, $C \sqcap D$, and $\neg C$ are concepts;
3. If C is a concept and R is a role, then $\forall R.C$ and $\exists R.C$ are concepts.

For example, suppose *Doctor, Man* are the given atomic concepts, *hasChild* is an atomic role, and *lucy, bill* are two individuals. Then, *Doctor* $\sqcap$ *Man* is a complex concept representing male doctors; the complex concept $\forall hasChild.Doctor$ means the concept representing things whose children are all doctors, and $\exists hasChild.\top$ is a complex concept corresponding to the set of individuals who have at least one child. Then $(\textit{Doctor} \sqcap \textit{Man})(bill)$ means that $bill$ is a male doctor, and $hasChild(lucy, bill)$ means that $bill$ is a child of $lucy$.

The formal definition of the semantics of $\mathcal{ALC}$ is given by means of interpretations $I = (\Delta^I, \cdot^I)$ consisting of a non-empty domain Δ^I and a mapping $\cdot^I$ satisfying the conditions in Table 1 whose third column is the translation $\Phi(C,x)$ ([14,13]) from every concept C to a first-order formula, where a, b are constant symbols, y is a fresh variable symbol and x is either the constant symbol a or a variable symbol. The unique name assumption is adapted by Description Logics.

An $\mathcal{ALC}$ knowledge base (or knowledge base for simplicity) consists of a set of assertions, called the $ABox$, and a set of inclusion axioms, calld the $TBox$. Assertions are of the form $C(a)$ or $R(a,b)$, where a, b are individuals and C and R are concepts and roles, respectively. Inclusion axioms are of the form $C \sqsubseteq D$, where C and D are concepts. The translation $\Phi(\cdot)$ from an axiom to a FOL formula is also given in the

Table 2. Semantics of $\mathcal{ALC}4$ Concepts

Constructor Syntax	Semantics
A	$A^I = \langle P, N\rangle$, where $P, N \subseteq \Delta^I$
R	$R^I = \langle R_P, R_N\rangle$, where $R_P, R_N \subseteq \Delta^I \times \Delta^I$
o	$o^I \in \Delta^I$
$C_1 \sqcap C_2$	$\langle P_1 \cap P_2, N_1 \cup N_2\rangle$, if $C_i^I = \langle P_i, N_i\rangle$ for $i = 1, 2$
$C_1 \sqcup C_2$	$\langle P_1 \cup P_2, N_1 \cap N_2\rangle$, if $C_i^I = \langle P_i, N_i\rangle$ for $i = 1, 2$
$\neg C$	$(\neg C)^I = \langle N, P\rangle$, if $C^I = \langle P, N\rangle$
$\exists R.C$	$\langle\{x \mid \exists y, (x, y) \in \mathit{proj}^+(R^I)$ and $y \in \mathit{proj}^+(C^I)\}$, $\{x \mid \forall y, (x, y) \in \mathit{proj}^+(R^I)$ implies $y \in \mathit{proj}^-(C^I)\}\rangle$
$\forall R.C$	$\langle\{x \mid \forall y, (x, y) \in \mathit{proj}^+(R^I)$ implies $y \in \mathit{proj}^+(C^I)\}$, $\{x \mid \exists y, (x, y) \in \mathit{proj}^+(R^I)$ and $y \in \mathit{proj}^-(C^I)\}\rangle$

third column of Table 1. Informally, an assertion $C(a)$ means that the individual a is an instance of the concept C, and an assertion $R(a, b)$ means that the individual a is related with the individual b via the property R. The inclusion axiom $C \sqsubseteq D$ means that each individual of C is an individual of D.

An interpretation satisfies an $\mathcal{ALC}$ knowledge base (i.e. is a model of the knowledge base) iff it satisfies each axiom in both the $ABox$ and the $TBox$. An $\mathcal{ALC}$ knowledge base is called satisfiable (unsatisfiable) iff there exists (does not exist) such a model. In $\mathcal{ALC}$, reasoning tasks, i.e. the derivation of logical consequences, can be reduced to satisfiability checking of ontologies [12,15].

From the translations from $\mathcal{ALC}$ axioms to FOL formulae shown in Table 1, $\mathcal{ALC}$ is a subset of FOL, which is proven decidable [12].

2.2 Four-Valued Semantics for $\mathcal{ALC}$

We consider the four-valued semantics for $\mathcal{ALC}$ given in [2]. Semantically, four-valued interpretations map individuals to elements of the domain of the interpretation, as usual. For concepts, however, to allow for reasoning with inconsistencies, a four-valued interpretation over a domain Δ^I assigns to each concept C a pair $\langle P, N\rangle$ of (not necessarily disjoint) subsets of Δ^I. Intuitively, P is the set of elements known to belong to the extension of C, while N is the set of elements known to be not contained in the extension of C. P and N are not necessarily disjoint and mutually complemental with respect to the domain.

Formally, a four-valued interpretation is a pair $I = (\Delta^I, \cdot^I)$ with Δ^I as domain, where $\cdot^I$ is a function assigning elements of Δ^I to individuals, and subsets of $(\Delta^I)^2$ to concepts, such that the conditions in Table 2 are satisfied, where functions $\mathit{proj}^+(\cdot)$ and $\mathit{proj}^-(\cdot)$ are defined by $\mathit{proj}^+\langle P, N\rangle = P$ and $\mathit{proj}^-\langle P, N\rangle = N$.

The idea of four-valued semantics is based on the idea of having four truth values, instead of the classical two. The four truth values stand for *true, false, unknown* and *contradictory*. We use the symbols $t, f, \ddot{\bot}, \ddot{\top}$, respectively, for these truth values, and the set of these four truth values is denoted by FOUR. The correspondence between truth values from FOUR and concept extensions are defined as follows:

Table 3. Semantics of inclusion axioms in $\mathcal{ALC}4$

Axiom Name	Syntax	Semantics
material inclusion	$C_1 \mapsto C_2$	$\Delta^I \setminus proj^-(C_1^I) \subseteq proj^+(C_2^I)$
internal inclusion	$C_1 \sqsubset C_2$	$proj^+(C_1^I) \subseteq proj^+(C_2^I)$
strong inclusion	$C_1 \rightarrow C_2$	$proj^+(C_1^I) \subseteq proj^+(C_2^I)$ and $proj^-(C_2^I) \subseteq proj^-(C_1^I)$
concept assertion	$C(a)$	$a^I \in proj^+(C^I)$
role assertion	$R(a,b)$	$(a^I, b^I) \in proj^+(R^I)$

Definition 1. *For instances $a \in \Delta^I$ and concept names C,*

- $C^I(a) = t$, *iff* $a^I \in proj^+(C^I)$ *and* $a^I \notin proj^-(C^I)$,
- $C^I(a) = f$, *iff* $a^I \notin proj^+(C^I)$ *and* $a^I \in proj^-(C^I)$,
- $C^I(a) = \ddot{\top}$, *iff* $a^I \in proj^+(C^I)$ *and* $a^I \in proj^-(C^I)$,
- $C^I(a) = \ddot{\bot}$, *iff* $a^I \notin proj^+(C^I)$ *and* $a^I \notin proj^-(C^I)$.

The correspondence between FOUR *and role extensions can be defined in a similar way.*

Obviously, for the semantics defined above, we ensure that a number of useful equivalences from classical DLs, such as the double negation law and the de Morgan Laws, hold.

The increase of truth values for four-valued semantics allows for several ways to define meaningful notions of four-valued implication. Indeed, there are three major notions of implication in the literature [16]. Corresponding to them, we have three ways to explain class inclusions in $\mathcal{ALC}$: the material inclusion axiom, the internal inclusion axiom, and the strong inclusion axiom, denoted as $C \mapsto D$, $C \sqsubset D$, and $C \rightarrow D$, respectively, to distinguish from classical class inclusion $C \sqsubseteq D$. The semantics of the three different types of inclusion axioms is formally defined in Table 3 (together with the semantics of concept assertions).

These three class inclusion axioms provide knowledge base engineers with a flexible way to define different knowledge bases according to their different semantics [2]. However, when 4-valued models are used to measure inconsistency, we will point out that in Section 4.1 only one of them, the material inclusion, is proper. This is also a reason why other 4-valued description logics [17,18] are not suitable for measuring inconsistency.

We say that a four-valued interpretation I satisfies a four-valued knowledge base KB (i.e. is a model of it) iff it satisfies each assertion and each inclusion axiom in KB. A knowledge base KB is 4-valued satisfiable (unsatisfiable) iff there exists (does not exist) such a model.

3 Domain-Dependent Inconsistency

In this section, we define a domain-dependent inconsistency in DLs. We first recall the notion of inconsistency in DLs.

Definition 2. *A knowledge base KB is classically inconsistent iff KB has no classical model. A knowledge base which is classically inconsistent is called an inconsistent knowledge base. Otherwise, it is called a consistent knowledge base.*

According to Definition 2, *KB* is inconsistent iff it has no classical model. However, given a knowledge base which is consistent, it may be "inconsistent" for a domain.

Example 3. *Given a knowledge base KB* $= \{\mathcal{T}, \mathcal{A}\}$*, where* $\mathcal{T} = \{A \sqsubseteq \exists R.\neg A\}$ *and* $\mathcal{A} = \{A(a)\}$*, KB is consistent because* KB *has a classical model* $I = \langle \Delta^I, \cdot^I \rangle$*, where* $\Delta^I = \{a, b\}$ *and* $A^I = \{a\}, R^I = \{(a, b)\}$*. However, KB has no classical model with respect to the domain* $\{a\}$*.*

We have the following definition of domain-dependent inconsistency.

Definition 4. *For a given domain* $\mathcal{D}$*, we call KB domain-dependently inconsistent with respect to* $\mathcal{D}$*, denoted* $\mathcal{D}$*-inconsistent, if KB has no classical model with respect to* $\mathcal{D}$*. Otherwise it is called* $\mathcal{D}$*-consistent.*

Example 5. *(Example 3 continued) Consider two domains* $\Delta^{I_1} = \{a\}$ *and* $\Delta^{I_2} = \{a, b\}$*. It is easy to check that KB is* Δ^{I_1}*-inconsistent, but* Δ^{I_2}*-consistent.*

Given another knowledge base $KB' = \{A \sqsubseteq \exists R.A, A(a)\}$, KB' is both Δ^{I_1}-consistent and Δ^{I_2}-consistent. Therefore, *KB* is "more inconsistent" than KB'. In the settings where only finite domains are considered, such as databases, the concept of domain-dependent (in)consistency can provide us with an approach to distinguish the extent of inconsistency of two logically consistent knowledge bases.

We give an important property of domain-dependent (in)consistency.

Proposition 6. *An* $\mathcal{ALC}$ *knowledge base KB is consistent, if and only if there exists a positive integer* N*, such that for any finite domain* $\mathcal{D}$ *whose cardinality is greater than* N *(i.e.* $|\mathcal{D}| \geq N$*), KB is* $\mathcal{D}$*-consistent.*

The proposition holds because of the finite model property of $\mathcal{ALC}$ and the fact that it is equality-free. It says that for a consistent knowledge base in $\mathcal{ALC}$, it will be domain-dependently consistent after the domain's cardinality becomes greater than a finite positive integer. Obviously, this property does not hold for other DLs in general and neither for FOL theories which do not have the finite model property or are not equality-free.

4 Inconsistency Measure

In this section, we measure inconsistency of an $\mathcal{ALC}$ knowledge base using four-valued models. In section 4.1 we discuss which kind of semantics of class inclusions as defined in Table 3 is appropriate to be used for measuring inconsistency. In section 4.2, we define the (domain-dependent) inconsistency degree of a knowledge base and study specially the properties of the inconsistency degree for $\mathcal{ALC}$ knowledge bases. Finally, in section 4.3, we give an ordering on knowledge bases based on the inconsistency degrees.

4.1 The Choice of Class Inclusion Axioms

Without explicit declaration, if a class inclusion axiom is expressed in the form $C \sqsubseteq D$, its semantics is the classical semantics as defined in Table 1. If it is in the form of $C \mapsto D, C \sqsubset D$, or $C \rightarrow D$, it is interpreted under the four-valued semantics as defined in Table 3.

Example 7. *Consider $\mathcal{T} = \{A \sqcup \neg A \sqsubseteq A \sqcap \neg A\}$ which is a TBox of an inconsistent knowledge base. Based on four-valued semantics, we have the following three ways to interpret the subsumption: $\mathcal{T}_1 = \{A \sqcup \neg A \mapsto A \sqcap \neg A\}$, $\mathcal{T}_2 = \{A \sqcup \neg A \sqsubset A \sqcap \neg A\}$, and $\mathcal{T}_3 = \{A \sqcup \neg A \rightarrow A \sqcap \neg A\}$, respectively. Now consider the following two 4-valued interpretations:*

$$I_1 = (\Delta^{I_1}, \cdot^{I_1}) : \ A^{I_1} = \langle \Delta^{I_1}, \Delta^{I_1} \rangle$$
$$I_2 = (\Delta^{I_2}, \cdot^{I_2}) : \ A^{I_2} = \langle \emptyset, \emptyset \rangle$$

According to Table 3, $\mathcal{T}_1$ has a unique 4-valued model I_1, while $\mathcal{T}_2$ and $\mathcal{T}_3$ both have I_1 and I_2 as 4-valued models.

In the above example, the difference between I_1 and I_2 is that I_1 assigns contradiction to the concept A, while I_2 assigns nothing to a contradictory value, though knowledge base $\mathcal{T}$ is inconsistent. Therefore, if we interpret a subsumption of an inconsistent knowledge base as internal or strong class inclusion axiom, there may exist a 4-valued model which does not assign contradiction to any concept or role name. We give a proposition which shows an important property of material inclusion. We first introduce some denotations.

Definition 8. *Let I be a four-valued model of KB with domain Δ^I, and let $\mathcal{L}_{KB}$ be the set of atomic concepts and roles occurring in KB. The* inconsistency set *of I for KB, written ConflictOnto(I, KB), is defined as follows:*

$$\mathit{ConflictOnto}(I, KB) = \mathit{ConflictConcepts}(I, KB) \cup \mathit{ConflictRoles}(I, KB),$$

where $\mathit{ConflictConcepts}(I, KB) = \{A(a) \mid A^I(a) = \ddot{\top}, A \in \mathcal{L}_{KB}, a \in \Delta^I\}$, and $\mathit{ConflictRoles}(I, KB) = \{R(a_1, a_2) \mid R^I(a_1, a_2) = \ddot{\top}, R \in \mathcal{L}_{KB}, a_1, a_2 \in \Delta^I\}$.

Intuitively, *ConflictOnto*(I, KB) is the set of conflicting atomic individual assertions.

Proposition 9. *Given an $\mathcal{ALC}$ knowledge base KB=$(\mathcal{T}, \mathcal{A})$, KB is inconsistent if and only if ConflictOnto$(I, KB) \neq \emptyset$ for very 4-valued model I of KB, provided that all class inclusion axioms in $\mathcal{T}$ are explained as material inclusions.*

According to Proposition 9 and the counterexample 7, it is more desirable to interpret class inclusion by material inclusion. So, in the rest of this section, we choose only the semantics of material inclusion as the 4-valued semantics of class inclusion. That is, other semantics are not used to measure inconsistency of an an knowledge base, though they are used to reason with an knowledge base in [2].

4.2 Inconsistency Degree

In this section, we give formal definitions of the inconsistency degree of an inconsistent knowledge base and the domain-dependent inconsistency degree for a consistent knowledge base.To do this, we need the following notions.

Definition 10. *For the knowledge base KB and a 4-valued interpretation I,*

$$\mathit{GroundOnto}(I, \mathit{KB}) = \mathit{GroundConcepts}(I, \mathit{KB}) \cup \mathit{GroundRoles}(I, \mathit{KB}),$$

where $\mathit{GroundConcepts}(I, \mathit{KB}) = \{A(a) \mid a \in \Delta^I, A \in \mathcal{L}_{\mathit{KB}}\}$, $\mathit{GroundRoles}(I, \mathit{KB}) = \{R(a_1, a_2) \mid a_1, a_2 \in \Delta^I, R \in \mathcal{L}_{\mathit{KB}}\}$.

Intuitively, GroundOnto(I, KB) is the collection of all atomic individual assertions.

In order to define the degree of inconsistency, we consider only interpretations with finite domains. This is reasonable in practical cases because only a finite number of individuals can be represented or would be used. This is also reasonable from the theoretical aspect because $\mathcal{ALC}$ has the finite model property — that is, if a knowledge base is consistent and within the expressivity of $\mathcal{ALC}$, then it has a classical model whose domain is finite.

Definition 11. *The* inconsistency degree *of a knowledge base w.r.t. a model $I \in \mathcal{M}4$ (KB), denote $\mathit{Inc}_I(\mathit{KB})$, is a value in $[0, 1]$ calculated in the following way:*

$$\mathit{Inc}_I(\mathit{KB}) = \frac{|\mathit{ConflictOnto}(I, \mathit{KB})|}{|\mathit{GroundOnto}(I, \mathit{KB})|}$$

That is, the inconsistency degree of *KB* w.r.t. I is the ratio of the number of conflicting atomic individual assertions divided by the amount of all possible atomic individual assertions of *KB* w.r.t. I. It measures to what extent a given knowledge base contains inconsistency w.r.t. I.

Example 12. *Consider knowledge base $\mathit{KB}_1 = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T} = \{A \sqsubseteq B \sqcap \neg B\}$, $\mathcal{A} = \{A(a)\}$. A 4-valued model of KB_1 is as follows: $I_1 = (\Delta^{I_1}, \cdot^{I_1})$, where $\Delta^{I_1} = \{a\}$, $A^{I_1}(a) = t$, and $B^{I_1}(a) = \ddot{\top}$. For this model, $\mathit{GroundOnto}(I_1, \mathit{KB}_1) = \{A(a), B(a)\}$, and $B(a)$ is the unique element in $\mathit{ConflictOnto}(I_1, \mathit{KB}_1)$. Therefore, $\mathit{Inc}_{I_1}(\mathit{KB}_1) = \frac{1}{2}$.*

In [11], it has been shown that for a fixed domain, not all the models need to be considered to define an inconsistency measure because some of them may overestimate the degree of inconsistency. Let us go back to Example 12.

Example 13. *(Example 12 Continued) Consider another 4-valued model of KB_1: $I_2 = (\Delta^{I_2}, \cdot^{I_2})$, where $\Delta^{I_2} = \{a\}$, $A^{I_2}(a) = \ddot{\top}$, $B^{I_2}(a) = \ddot{\top}$. I_1 and I_2 share the same domain. Since $|\mathit{ConflictOnto}(I_2, \mathit{KB}_1)| = |\{B(a), A(a)\}| = 2$, we have $I_1 \leq_{\mathit{Incons}} I_2$ by Definition 14. This is because $\cdot^{I_2}$ assigns contradiction to $A(a)$. However, $A(a)$ is not necessary a conflicting axiom in four-valued semantics. Therefore, we conclude that $\mathit{Inc}_{I_2}(\mathit{KB})$ overestimates the degree of inconsistency of KB_1.*

We next define a partial ordering on $\mathcal{M}4(\mathit{KB})$ such that the minimal elements w.r.t. it can be used to define the inconsistency measure for *KB*.

Definition 14. *(Model ordering w.r.t. inconsistency) Let I_1 and I_2 be two four-valued models of a knowledge base KB such that $|\Delta_1^I| = |\Delta_2^I|$. We say the inconsistency of I_1 is less than or equal to I_2, written $I_1 \leq_{Incons} I_2$, if and only if $Inc_{I_1}(KB) \leq Inc_{I_2}(KB)$.*

The condition $|\Delta^{I_1}| = |\Delta^{I_2}|$ in this definition just reflects the perspective that only models with the same cardinality of domain are comparative. As usual, $I_1 <_{Incons} I_2$ denotes $I_1 \leq_{Incons} I_2$ and $I_2 \not\leq_{Incons} I_1$, and $I_1 \equiv_{Incons} I_2$ denotes $I_1 \leq_{Incons} I_2$ and $I_2 \leq_{Incons} I_1$. $I_1 \leq_{Incons} I_2$ means that I_1 is more consistent than I_2.

The model ordering w.r.t. inconsistency is used to define preferred models.

Definition 15. *Let KB be a DL-based knowledge base and $n(n \geq 1)$ be a given cardinality. The* preferred models w.r.t $\leq_{Incons}$ of size n*, written $PreferModel_n(KB)$, are defined as follows:*

$$PreferModel_n(KB) = \{I \mid |\Delta^I| = n; \forall I' \in \mathcal{M}4(KB), |\Delta^{I'}| = n \text{ implies } I \leq_{Incons} I'\}$$

That is, $PreferModel_n(KB)$ is the set of all models of size n which are minimal with respect to $\leq_{Incons}$.

From Definition 14 and Definition 15, it is easy to see that for the preferred models I_1 and I_2 with a same cardinality, inconsistency degrees of the knowledge base $w.r.t$ them are equal. That is, $Inc_{I_1}(KB) = Inc_{I_2}(KB)$, which means I_1 and I_2 have the same amount of contradictory atomic assertions, though the elements of their domains may be quite different.

For simplicity, we say an interpretation is *well-sized* if and only if the cardinality of its domain is equal to or greater than the number of individuals in *KB*. Because of the unique name assumption of the DL $\mathcal{ALC}$, an interpretation can be a model only if it is well-sized. Moreover, the following theorem asserts the existence of preferred models among the well-sized interpretations.

Theorem 16 *For any given $\mathcal{ALC}$ knowledge base KB, preferred models among well-sized interpretations always exist.*

Above we have considered the inconsistency degrees of knowledge bases with respect to four-valued models, especially with respect to preferred models. Now we define an integrated inconsistency degree of a knowledge base allowing for different domains.

Definition 17. *Given a knowledge base KB and an arbitrary cardinality $n(n \geq 1)$, let I_n be an arbitrary model in $PreferModels_n(KB)$. The inconsistency degree sequence of KB, called OntoInc(KB), is defined as $\langle r_1, r_2, ..., r_n, ...\rangle$, where $r_n = Inc(I_n, KB)$ if I_n is well-sized. and let $r_n = *$ otherwise.*

In Definition 17, we use $*$ as a kind of null value. Given a domain with size n, we have the following three cases: (1) if $r_n = *$, it means that the knowledge knowledge base has no 4-valued models with size n; (2)if $r_n = 0$, it means that KB has a classical model among its 4-valued models. (3) if $r_n > 0 (\neq *)$, it means KB has no classical models but has 4-valued models.

From Theorem 16 and the unique name assumption of $\mathcal{ALC}$ the following property holds obviously.

Proposition 18. *Assume KB is a knowledge base and* $OntoInc(KB) = \langle r_1, r_2, ...\rangle$, *and* N *is the number of individuals of KB. Then*

$$r_i \begin{cases} = * & \text{if } 0 < i < N, \\ \geq 0 (\neq *) & \text{if } i \geq N. \end{cases}$$

This proposition shows that for a knowledge base, its inconsistency measure cannot be a meaningless sequence — that is, each element is the null value $*$. Moreover, the non-null values in the sequence start just from the position which equals the number of individuals in the knowledge base, and remains greater than zero in the latter positions of the sequence for inconsistent knowledge bases and becomes zero after n becomes large enough for consistent $\mathcal{ALC}$ knowledge bases.

Example 19. *(Example 7 continued) Obviously, for any four-valued model* $I = \langle \Delta^I, \cdot^I \rangle$ *of* $\mathcal{T}$, A *must be assigned to* $\langle \Delta^I, \Delta^I \rangle$, *therefore OntoInc(KB)* $= \{1, 1, ...\}$.

Example 20. *(Example 12 continued) Each preferred model I of* KB_1 *must satisfy that (1) it assigns one and only one individual assertion in* $\{B(a), A(a)\}$ *to the contradictory truth value* $\ddot{\top}$ *— that is,* $B^I(a) = \ddot{\top}$ *and* $A(a) = t$, *or* $B^I(a) = t$ *and* $A(a) = \ddot{\top}$; *(2) it assigns other grounded assertions to truth values among the set* $\{t, f, \ddot{\bot}\}$. *So* $|ConflictOnto(I, KB)| = 1$. *Consequently,* $OntoInc(KB_1) = \{\frac{1}{2}, \frac{1}{4}, ..., \frac{1}{2n}, ...\}$.

Example 21. *(Example 5 continued) It is easy to check that OntoInc(*KB'*)* $= \langle 1, 0, 0, ...\rangle$, *while OntoInc(*$KB''$*)* $= \langle 0, 0, ...\rangle$. *By definition 22,* KB' *has more domain-dependent inconsistency than* KB'', *that is,* $KB'' \prec_{Incons} KB'$.

4.3 Ordering Knowledge Bases with Respect to Their Inconsistent Degree

In this section, we define an ordering over all knowledge bases inspired by [19].

Definition 22. *Given two knowledge bases* KB_1 *and* KB_2, *assume* $OntoInc(KB_1) = \langle r_1, r_2, ...\rangle$ *and* $OntoInc(KB_2) = \langle r'_1, r'_2, ...\rangle$. *We say* KB_1 *is strictly less inconsistent than* KB_2, *written* $KB_1 \prec_{Incons} KB_2$, *iff one of the following conditions holds:*

1. $N_1 < N_2$
2. $N_1 = N_2$, $r_n \leq r'_n (\forall n \geq K)$, *and there exists* $n_0 \geq K$ *such that* $r_n < r'_n$

where $N_1 = min\{i : r_i = 0\}, N_2 = min\{i : r'_i = 0\}, K = min\{i : r_i \neq *, r'_i \neq *\}$.

N_1 (or N_2) is the first position from which the elements of the sequence $\langle r_1, r_2, ...\rangle$ (or $\langle r'_1, r'_2, ...\rangle$) become 0. For an inconsistent knowledge base, the position is infinite, so we denote $N_1 = \infty$ which is strictly greater than any finite number. So does N_2. According to Proposition 18, $\prec_{Incons}$ is well-defined. Moreover, an equality of two knowledge bases can be defined by $KB_1 =_{Incons} KB_2$ if and only if for all $n \geq K, r_n = r'_n$.

By Definition 22 and Proposition 6, obviously, all consistent knowledge bases are less inconsistent than any inconsistent knowledge base.

For consistent knowledge bases, according to Definition 22, one knowledge base is less domain-inconsistent than the other if and only if 0 begins at an earlier position in

its inconsistency degree sequence than that in the inconsistency degree sequence of the other knowledge base.

For the ordering among inconsistent knowledge bases, we only compare the values from the position at which both sequences have non-null values, according to Definition 22. This is because there exist infinite elements of sequences of their inconsistency degree which are non-null, and non-zero. These elements together are to reflect the useful information about the inconsistency of knowledge bases.

Example 23. *(Example 20 continued) Suppose* $KB_2 = \{A \sqsubseteq B \sqcap \neg B, A \sqsubseteq C, A(a)\}$. *In its preferred models, the individual assertions related to* C *are not involved with the contradictory truth value, so* $OntoInc(KB_2) = \{\frac{1}{3}, \frac{1}{6}, ..., \frac{1}{3n}, ...\}$. *By definition 22,* $KB_2 \prec_{Incons} KB_1$, *which means that* KB_2 *is less inconsistent than* KB_2.

Example 24. *(Example 19, 21, 23 continued)*

$$KB'' \prec_{Incons} KB' \prec_{Incons} KB_2 \prec_{Incons} KB_1 \prec_{Incons} KB.$$

5 Related Work and Conclusion

This paper provides a way to distinguish description logic based knowledge bases considering their different inconsistency degrees.

In the literature, there are basically two other works on defining four-valued semantics for description logics [17,18]. However, their definitions of class inclusion axioms are actually the same as the internal inclusion defined in Table 3, so that their approaches are not suitable for measuring inconsistency according to our analysis in Section 4.1.

Our work is closely related to the work of inconsistency measuring given in [11], where Quasi-Classical models (QC logic [20]) are used as the underlying semantics. In this paper, we use four-valued models for description logics as the underlying semantics. This is because QC logic needs to translate each formula in the theory into prenex conjunctive normal form (PCNF). This is not practical, especially for a large knowledge base, because it may be quite time consuming and users probably do not like their knowledge bases to be modified syntactically. In this paper, we can see that four-valued models also provide us with a novel way to distinguish knowledge bases with different inconsistency degrees.

It is apparent that the inconsistency measure defined by our approach can be used to compute each axiom's contribution to inconsistency of a whole knowledge base by adapting the method proposed in [8], thereby providing important information for resolving inconsistency in a knowledge base. Moreover, we find that four-valued models may provide us with some way to quantify also the incompleteness degree of knowledge bases because of the additional truth value $\ddot{\bot}$ with respect to three-valued semantics, which is among our future work.

In [11], every set of formulae definitely has at least one QC model because neither the constant predicate t (tautology) nor the constant predicate f (false) is contained in the language. However, corresponding to t and f, the top concept $\top$ and bottom concept $\bot$ are two basic concept constructors for $\mathcal{ALC}$. Due to space limitation, we presume that the ontologies do not use $\top$ and $\bot$ as concept constructors. The discussion of the inconsistency measure for an arbitrary inconsistent ontology will be left as future work.

For an implementation of our approach, the key point is to compute the number of conflicting assertions in a preferred model with respect to any given finite domain. We are currently working on the algorithm, which will be presented in a future paper.

References

1. Huang, Z., van Harmelen, F., ten Teije, A.: Reasoning with inconsistent ontologies. In: Proc. of IJCAI'05, Professional Book Center (2005)
2. Ma, Y., Hitzler, P., Lin, Z.: Algorithms for paraconsistent reasoning with OWL. In: Proc. of ESWC'07, pp. 399–413 (2007)
3. Schlobach, S.: Diagnosing terminologies. In: Proc. of AAAI'05, pp. 670–675 (2005)
4. Parsia, B., Sirin, E., Kalyanpur, A.: Debugging OWL ontologies. In: Proc. of WWW'05, pp. 633–640 (2005)
5. Hunter, A.: How to act on inconsistent news: Ignore, resolve, or reject. Data Knowl. Eng. 57, 221–239 (2006)
6. Hunter, A.: Measuring inconsistency in knowledge via quasi-classical models. In: Proc. of AAAI/IAAI, pp. 68–73 (2002)
7. Hunter, A., Konieczny, S.: Approaches to measuring inconsistent information. In: Bertossi, L., Hunter, A., Schaub, T. (eds.) Inconsistency Tolerance. LNCS, vol. 3300, pp. 191–236. Springer, Heidelberg (2005)
8. Hunter, A., Konieczny, S.: Shapley inconsistency values. In: Proc. of KR'06, pp. 249–259 (2006)
9. Knight, K.: Measuring inconsistency. Journal of Philosophical Logic 31, 77–98 (2001)
10. Oller, C.A.: Measuring coherence using LP-models. J. Applied Logic 2, 451–455 (2004)
11. Grant, J., Hunter, A.: Measuring inconsistency in knowledgebases. J. Intell. Inf. Syst. 27, 159–184 (2006)
12. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
13. Borgida, A.: On the relative expressiveness of description logics and predicate logics. Artif. Intell. 82, 353–367 (1996)
14. Schaerf, M., Cadoli, M.: Tractable reasoning via approximation. Artif. Intell. 74, 249–310 (1995)
15. Horrocks, I., Patel-Schneider, P.F.: Reducing OWL entailment to description logic satisfiability. J. Web Sem. 1, 345–357 (2004)
16. Arieli, O., Avron, A.: The value of the four values. Artif. Intell. 102, 97–141 (1998)
17. Patel-Schneider, P.F.: A four-valued semantics for terminological logics. Artif. Intell. 38, 319–351 (1989)
18. Straccia, U.: A sequent calculus for reasoning in four-valued description logics. In: Galmiche, D. (ed.) TABLEAUX 1997. LNCS, vol. 1227, pp. 343–357. Springer, Heidelberg (1997)
19. Grant, J.: Classifications for inconsistent theories. Notre Dame J. Formal Logic 19, 435–444 (1978)
20. Hunter, A.: A semantic tableau version of first-order quasi-classical logic. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, pp. 544–555. Springer, Heidelberg (2001)

On the Dynamics of Total Preorders: Revising Abstract Interval Orders

Richard Booth[1] and Thomas Meyer[2]

[1] Mahasarakham University, Faculty of Informatics, Mahasarakham 44150, Thailand
richard.b@msu.ac.th
[2] Meraka Institute, CSIR, Pretoria, South Africa
tmeyer@csir.co.za

Abstract. Total preorders (tpos) are often used in belief revision to encode an agent's strategy for revising its belief set in response to new information. Thus the problem of tpo-revision is of critical importance to the problem of iterated belief revision. Booth et al. [1] provide a useful framework for revising tpos by adding extra structure to guide the revision of the initial tpo, but this results in *single-step* tpo revision only. In this paper we extend that framework to consider *double-step* tpo revision. We provide new ways of representing the structure required to revise a tpo, based on *abstract interval orders*, and look at some desirable properties for revising this structure. We prove the consistency of these properties by giving a concrete operator satisfying all of them.

1 Introduction

Total preorders (*tpos* for short) are used to represent preferences in many contexts. In particular, in the area of belief revision [2], a common way to encode an agent's *strategy* for revising its belief set is via a tpo $\leq$ over the set W of possible worlds [3,4]. The agent's current belief set is identified with the set of sentences true in all the most preferred worlds, while upon receiving new evidence α, its *new* belief set is calculated with the help of $\leq$, typically by taking it to be the set of sentences true in all the most preferred worlds in which α holds. Of course in order to enable a *further* revision, what is needed is not just a new belief set, but also a new *tpo* to go with it. Thus the problem of *tpo-revision* is of critical importance to the problem of *iterated belief revision* [5,6,7].

In the problem of *belief set* revision, the tpo $\leq$ can be thought of as *extra structure* which is brought in to guide revision of the belief set. This extra structure goes *beyond* that given by the initial belief set, in the sense that the belief set can be extracted from it. Thus one natural way to attack the problem of tpo revision is to call up even more extra structure, let's denote it by X, which similarly goes beyond $\leq$ and can be used to guide revision of $\leq$. This is the approach taken by Booth et al. [1] where X takes the form of a purely qualitative structure (to be described in more detail below). Other, more quantitative forms are also conceivable [8]. Either way, X is used to determine a revised tpo $\leq^*_\alpha$ given any new evidence α. However, there is a problem with this approach regarding

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 42–53, 2007.

iterated tpo-revision: While the extra structure X tells us how to determine a new tpo $\leq^*_\alpha$, it tells us nothing about how to determine the *new* extra structure X^*_α to go with $\leq^*_\alpha$ which can then guide the *next* revision. Clearly the problem of iterated belief revision has simply re-emerged "one level up". The purpose of this paper is to investigate this problem in the particular case when the extra structure X takes the form studied by Booth et al. [1].

The intuition behind the family of tpo-revision operators defined by Booth et al. is that *context* ought to play a role when comparing different possible worlds according to preference. The starting point is to assume that to each possible world x are associated two abstract objects x^+ and x^-. Intuitively, x^+ will represent x in contexts favourable to it, while x^- will be the representative of x in those contexts unfavourable to it. Then, along with the initial tpo $\leq$ over W to be revised, it is assumed an agent has a tpo $\preceq$ over this entire set of objects $W^\pm$. This new tpo $\preceq$ represents the additional structure X which is used to encode the agent's strategy for revising $\leq$ in response to new evidence α. The arrival of α is seen as a context favourable to (a "good day" for) those worlds consistent with α, and a context unfavourable to (a "bad day" for) for all the other worlds. Thus the revised tpo $\leq^*_\alpha$ is obtained by setting $x \leq^*_\alpha y$ iff $x^\epsilon \preceq y^\delta$, with the values $\epsilon, \delta \in \{+, -\}$ dependent on whether x, y satisfy α or not. As was shown by Booth et al. [1], the family of tpo-revision operators so generated is characterised exactly by a relatively small list of rules, including several well-known properties which have previously been proposed. The family also includes as special cases several specific, and diverse, operators which have previously been studied [7,9]. Thus, this framework constitutes an important contribution to single-step tpo-revision.

The plan of the paper is as follows. In Section 2 we recall the framework for single-step tpo-revision described by Booth et al. [1]. We give the formal definition of the orderings $\preceq$ described above and introduce a useful new graphical representation of these orderings in terms of *abstract intervals*. In Section 3 we introduce an alternative way of representing this structure which we call *strict preference hierarchies* (SPHs). We show that these are equivalent to the $\preceq$ orderings. A consequence of this is that the problem mentioned above of determining $\preceq^*_\alpha$ may be equivalently posed as the problem of revising SPHs. In Section 4 we consider a few desirable properties which any good operator for revising SPHs should satisfy, before proving the consistency of these properties in Section 5 by providing an example of a concrete operator which is shown to satisfy them all. We conclude and mention ideas for further research in Section 6.

Preliminaries: We work in a finitely-generated propositional language L. As mentioned above, the set of propositional worlds is denoted by W. Given a sentence $\alpha \in L$, $[\alpha]$ denotes the set of worlds which satisfy α. Classical logical equivalence over L is denoted by $\equiv$. A total preorder is any binary relation $\leq$ (or $\preceq$) which is transitive and connected. For any such relation $<$ (or $\prec$) denotes its strict part ($x < y$ iff both $x \leq y$ and $y \not\leq x$) and $\sim$ its symmetric closure ($x \sim y$ iff both $x \leq y$ and $y \leq x$). For each $\alpha \in L$ it will be useful to define the tpo $\leq^\alpha$ over W *generated by* α by setting $x \leq^\alpha y$ iff $x \in [\alpha]$ or $y \in [\neg\alpha]$.

2 Single-Step Revision of Tpos

We let $W^{\pm} = \{x^{\epsilon} \mid x \in W \text{ and } \epsilon \in \{+,-\}\}$, and we assume, for any $x, y \in W$ and $\epsilon, \delta \in \{+,-\}$, that $x^{\epsilon} = y^{\delta}$ only if both $x = y$ and $\epsilon = \delta$. In other words all these abstract objects are distinct. Then we assume a given order $\preceq$ over $W^{\pm}$ satisfying the following conditions:

($\preceq$1) $\preceq$ is a total preorder
($\preceq$2) $x^+ \preceq y^+$ iff $x^- \preceq y^-$
($\preceq$3) $x^+ \prec x^-$

Rule ($\preceq$2) was split by Booth et al. [1] into two separate rules "$x^+ \preceq y^+$ iff $x \leq y$" and "$x^- \preceq y^-$ iff $x \leq y$", which made reference to an explicitly given initial tpo $\leq$ over W which is meant to be revised. However we can clearly recover $\leq$ from $\preceq$ satisfying the above three rules. We just define it by $x \leq y$ iff $x^+ \preceq y^+$ (or $x \leq y$ iff $x^- \preceq y^-$). In this case we say $\leq$ is the tpo over W *associated to* $\preceq$, or that $\preceq$ is $\leq$*-faithful*. From this $\leq$ in turn we can if we wish extract the *belief set* associated to $\preceq$: it is the set of sentences true in all the minimal $\leq$-worlds. However in this paper the dynamics of the belief set is not so much the focus as that of $\leq$, or indeed $\preceq$.

How can we picture these orderings $\preceq$? One way was given by Booth et al. [1], using an assignment of numbers to a $2 \times n$ array, where n is the number of *ranks* according to the tpo associated to $\preceq$. In this paper we would like to suggest an alternative graphical representation which is perhaps more intuitive, and is easier to work with when trying to construct examples. The idea is, for each $x \in W$, to think of the pair (x^+, x^-) as representing an *abstract interval* assigned to x. We can imagine that to each x we assign a "stick" whose left and right endpoints are x^+ and x^- respectively. Condition ($\preceq$1) says the endpoints of all these possible sticks are totally preordered. ($\preceq$2) says the left endpoints of any two of these sticks always stand in exactly same relation to each other as the right endpoints, just as if all the sticks have the same length. ($\preceq$3) demands the stick-lengths are non-zero. We may arrange the sticks in an order such as the one shown in Figure 1, which shows the sticks associated to the five worlds x_1–x_5. The further to the left an endpoint is, the lower, i.e., more preferred, it is according to $\preceq$. Thus we see for example that $x_1^+ \prec x_3^+$ and $x_2^- \sim x_4^+$.

We are assuming the sticks as having equal length, but this is mainly for visual convenience. It has no semantic significance in the framework.

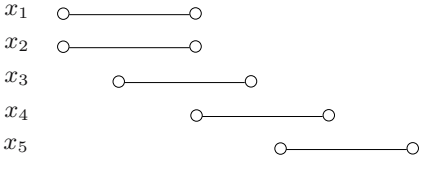


Fig. 1. Example interval ordering

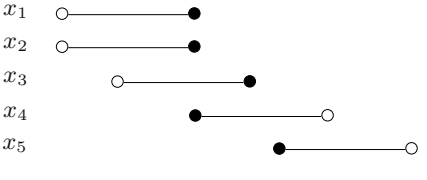


Fig. 2. Example revision

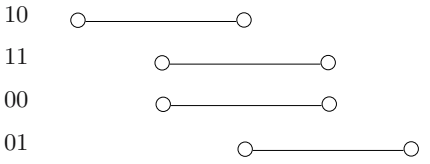


Fig. 3. Example interval ordering

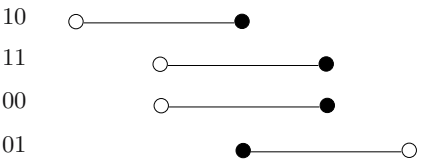


Fig. 4. Example revision

Given a tpo $\leq$ over W, we may use a $\leq$-faithful tpo $\preceq$ over $W^{\pm}$ to define a revision operator $*$ for $\leq$. The idea is that when evidence α arrives it casts a more favourable light on worlds satisfying α. So we consider α as signalling a "good" day for the α-worlds, and a "bad day" for the $\neg\alpha$-worlds. This leads us to define a new, revised tpo $\leq^*_\alpha$ by setting, for each $x, y \in W$, $x \leq^*_\alpha y$ iff $r_\alpha(x) \preceq r_\alpha(y)$, where, for any $x \in W$ and $\alpha \in L$,

$$r_\alpha(x) = \begin{cases} x^+ \text{ if } x \in [\alpha] \\ x^- \text{ if } x \in [\neg\alpha]. \end{cases}$$

In terms of our new picture, each world gets mapped to one of the endpoints of the stick associated to it – left if it is an α-world and right if it is a $\neg\alpha$-world. From this the new tpo $\leq^*_\alpha$ may be read off. For example in Figure 1 suppose we revise by α such that $x_4, x_5 \in [\alpha]$ and $x_1, x_2, x_3 \in [\neg\alpha]$. Then $\leq^*_\alpha$ may be read off by looking at the black circles in Figure 2. So we see $x_1 \sim^*_\alpha x_2 \sim^*_\alpha x_4 <^*_\alpha x_3 <^*_\alpha x_5$.

Example 1. For a more concrete example (recast from one in [1] which used the old graphical representation) we assume L is generated from just two variables p, q, leading to four worlds each of which we may denote as a pair of digits denoting the truth-values of p, q respectively. The sticks associated to each world are given in Fig. 3. The initial tpo $\leq$ is specified by $10 < 11 \sim 00 < 01$. Revising by $\neg p \wedge q$ leads to Fig. 4 from which we read off $10 \sim^*_{\neg p \wedge q} 01 <^*_{\neg p \wedge q} 11 \sim^*_{\neg p \wedge q} 00$.

If we look at the *belief set* associated to the new tpo $\leq^*_{\neg p \wedge q}$ in this example then we see it does not contain the new evidence $\neg p \wedge q$ due to the presence of world 10 among the minimal worlds in $\leq^*_{\neg p \wedge q}$. Thus we see that, at the level of belief sets, we are in the realm of so-called *non-prioritised* belief revision [10].

Given a fixed initial tpo $\leq$ over W, if the revision operator $*$ for $\leq$ can be defined from some $\leq$-faithful tpo $\preceq$ over $W^{\pm}$ as above then $*$ is said to be *generated by* $\preceq$. Booth et al. [1] characterised the class of revision operators for $\leq$ which can be generated from some $\preceq$. A revision operator $*$ can be generated from a $\leq$-faithful tpo over $W^{\pm}$ iff it satisfies the following properties for any $\alpha, \gamma \in L$:

($*1$) $\leq^*_\alpha$ is a tpo over W
($*2$) $\alpha \equiv \gamma$ implies $\leq^*_\alpha = \leq^*_\gamma$
($*3$) If $x, y \in [\alpha]$ then $x \leq^*_\alpha y$ iff $x \leq y$
($*4$) If $x, y \in [\neg\alpha]$ then $x \leq^*_\alpha y$ iff $x \leq y$
($*5$) If $x \in [\alpha]$, $y \in [\neg\alpha]$ and $x \leq y$ then $x <^*_\alpha y$
($*6$) If $x \in [\alpha]$, $y \in [\neg\alpha]$ and $y \leq^*_\alpha x$ then $y \leq^*_\gamma x$
($*7$) If $x \in [\alpha]$, $y \in [\neg\alpha]$ and $y <^*_\alpha x$ then $y <^*_\gamma x$

Rule ($*1$) just says revising a tpo over W should result in another tpo over W. ($*2$) is a syntax-irrelevance property. ($*3$) and ($*4$) are well-known as (CR1) and (CR2) [6]. They say the relative ordering of the α-worlds, respectively the $\neg\alpha$-worlds, should remain unchanged after receiving α. ($*5$) was introduced independently by Booth et al. [5] and Jin & Thielscher [11]. It says if an α-world x was considered at least as preferred as a $\neg\alpha$-world y *before* receiving α, then *after* receiving α, x should be *strictly* preferred to y. ($*6$) says that if a world x is not more preferred to a world y, *even after* receiving evidence α which clearly points more to x being the case than it does to y, then there can be *no* evidence which will lead to x being more preferred to y. ($*7$) is similar. Rule ($*2$) is actually redundant in this list, since it can be proved from the other rules [1].

3 Strict Preference Hierarchies

A given ordering $\preceq$ over $W^{\pm}$ satisfying ($\preceq 1$)–($\preceq 3$) represents the structure required to revise its associated tpo $\leq$ over W. In this section we introduce a way of re-packaging that structure. As observed by Booth et al. [1], from a single $\preceq$ we can extract *three* different notions of *strict preference* over W. First we have the simple one given by $x < y$ iff $x^+ \prec y^+$ (equivalently $x < y$ iff $x^- \prec y^-$), i.e., $<$ is just the strict part of the tpo over W associated to $\preceq$. In terms of our new graphical representation, $x < y$ iff the stick corresponding to x lies to the left of that associated to y, but *possibly with some overlap*. For example in Figure 1 we have $x_1 < x_3$. A second, stronger notion of strict preference can be expressed by: $x \lll y$ iff $x^- \prec y^+$. In other words, $x \lll y$ iff x, *even on a bad day*, is preferred to y or, in terms of the picture, iff the stick associated to x lies *completely to the left* of that associated to y, and furthermore there is "daylight" between them. E.g., in Figure 1 we see $x_2 \lll x_5$. Finally a third case, intermediate between $\lll$ and $<$, can be expressed by: $x \ll y$ iff $x^- \preceq y^+$. In other words $x \ll y$ iff x on a bad day is *at least as* preferred to y. This third case captures a "hesitation" [12] between strong strict preference $\lll$ and mere ordinary strict preference $<$. We will have $x \ll y$ and $x \not\lll y$ precisely when the right endpoint x^- of the x-stick and the left endpoint y^+ of the y-stick are vertically aligned with each other. E.g., in Figure 1 we have $x_1 \not\lll x_4$ but $x_1 \ll x_4$.We are now in a position to define our alternative representation of the structure used by Booth et al.

Definition 1. *The triple* $\mathbb{S} = (\lll, \ll, <)$ *of binary relations over* W *is a* strict preference hierarchy (over W) *(SPH for short) iff there is some relation* $\preceq$ *over*

$W^{\pm}$ satisfying (⪯1)–(⪯3) such that $\lll, \ll$ and $<$ can all be defined from $\preceq$ as above. We shall sometimes say that $\mathbb{S}$ *is* relative to $<$.

Such "interval orderings" like the above have already been studied in the context of temporal reasoning [13], as well as in preference modelling [12]. Indeed, concerning the former case, the relations $\lll, \ll, <$ could all be defined in terms of the relations *before*, *meets* and *overlaps* between temporal intervals studied by Allen [13].

What are the properties of the three relations $(\lll, \ll, <)$? A couple were already mentioned by Booth et al. [1]. For example we already know from there that $\lll$ and $\ll$ are strict partial orders (i.e., irreflexive and transitive). But what else do they satisfy? In particular how do they *interrelate* with each other? Furthermore, given any *arbitrary* triple $\mathbb{S} = (\lll, \ll, <)$ of binary relations over W, under what conditions on $\mathbb{S}$ can we be sure that $\mathbb{S}$ forms an SPH, i.e., under what conditions can we be sure there is *some* $\preceq$ satisfying (⪯1)–(⪯3) such that $\mathbb{S}$ can be derived from $\preceq$ in the above manner. These questions are answered by the following representation result for SPHs. We point out that part *(iii)* of the "only if" part (but not the "if" part) was essentially already proved, in the temporal reasoning context, by Allen [13].

Theorem 1. *Let $\lll, \ll$ and $<$ be three binary relations over W. Then $\mathbb{S} = (\lll, \ll, <)$ is an SPH iff the following conditions hold (where $x \leq y$ iff $y \not< x$):*
(i). *$\leq$ is a total preorder.*
(ii). $\lll \subseteq \ll \subseteq <$.
(iii). *The following are satisfied, for all $x, y, z \in W$:*

(SPH1) $z \leq x$ *and* $x \lll y$ *implies* $z \lll y$
(SPH2) $x \lll y$ *and* $y \leq z$ *implies* $x \lll z$
(SPH3) $z \leq x$ *and* $x \ll y$ *implies* $z \ll y$
(SPH4) $x \ll y$ *and* $y \leq z$ *implies* $x \ll z$
(SPH5) $z < x$ *and* $x \ll y$ *implies* $z \lll y$
(SPH6) $x \ll y$ *and* $y < z$ *implies* $x \lll z$

The rules (SPH1)–(SPH6) each represent some sort of transitivity condition across the relations of the SPH.

The "only if" direction of Theorem 1 is quite straightforward to prove, and in fact easy to visualise given our new graphical representation of $\preceq$. For the "if" direction, we may translate any triple $\mathbb{S} = (\lll, \ll, <)$ into a binary relation $\preceq_{\mathbb{S}}$ over $W^{\pm}$ as follows: Given $x^{\epsilon}, y^{\delta} \in W^{\pm}$, if $\epsilon = \delta$ then we set $x^{\epsilon} \preceq_{\mathbb{S}} y^{\epsilon}$ iff $x \leq y$. This ensures $\preceq_{\mathbb{S}}$ satisfies (⪯2). If $\epsilon \neq \delta$ but $x = y$ then we declare $x^{+} \prec_{\mathbb{S}} x^{-}$. This ensures (⪯3) is satisfied. Finally if $\epsilon \neq \delta$ and $x \neq y$ then we set $x^{+} \preceq_{\mathbb{S}} y^{-}$ iff $y \not\lll x$ and $x^{-} \preceq_{\mathbb{S}} y^{+}$ iff $x \ll y$. Then if $\mathbb{S}$ satisfies conditions *(i)–(iii)* from the theorem, then $\preceq_{\mathbb{S}}$ satisfies (⪯1) in addition to (⪯2) and (⪯3). Furthermore the SPH corresponding to $\preceq_{\mathbb{S}}$ is precisely $\mathbb{S}$ itself.

Two special limiting cases of SPHs were already mentioned by Booth et al. [1]: Given any tpo $\leq$ over W with strict part $<$, the triples $(\emptyset, \emptyset, <)$ and $(<, <, <)$ each *always* forms an SPH, as can easily be seen by checking conditions *(i)–(iii)* of the theorem. In fact these are the SPH forms of the well-known lexicographic

tpo-revision operator [7] and Papini's [9] "reverse" lexicographic tpo-revision operator respectively.

SPHs seem closely related to the notion of "PQI interval order" studied by Öztürk et al. [12]. Indeed several representation results in the same spirit as Theorem 1 can be found in their work. The main difference with ours is that PQI interval orders make use of an explicit numerical scale, so the endpoints of the intervals are ordinary real numbers, whereas our intervals are "abstract", having endpoints only in some totally preordered set (but see Section 5 of this paper). Also, with PQI interval orders, different possibilities (i.e., possible worlds for us) may be assigned intervals of different length. It is even possible for the interval assigned to one possibility to be completely *enclosed* in the interval assigned to another. This is something we do not allow. We are currently examining in more detail the relationship between SPHs and PQI interval orders.

To summarise the findings of this section, we now see we have two different, but equivalent ways of describing the structure required to revise a tpo $\leq$:

1. As a $\leq$-faithful tpo $\preceq$ over $W^{\pm}$ satisfying ($\preceq$1)–($\preceq$3).
2. As a triple $(\lll, \ll, <)$ of binary relations over W satisfying conditions *(i)–(iii)* from Theorem 1 (with $<$ being the strict part of $\leq$).

Recall that the revision operator $*$ for $\leq$ derived from a $\leq$-faithful tpo $\preceq$ over $W^{\pm}$ is defined by setting $x \leq^*_\alpha y$ iff $r_\alpha(x) \preceq r_\alpha(y)$. The next result shows how we can describe $*$ purely in terms of the SPH corresponding to $\preceq$.

Proposition 1. *Let $\leq$ be a tpo over W and let $\preceq$ be a given $\leq$-faithful tpo over $W^{\pm}$. Let $\mathbb{S} = (\lll, \ll, <)$ be the SPH corresponding to $\preceq$ and let $*$ be the revision operator for $\leq$ derived from $\preceq$. Then, for all $x, y \in W$,*

$$x \leq^*_\alpha y \text{ iff } \begin{cases} x \sim^\alpha y \text{ and } x \leq y \\ \text{or } x <^\alpha y \text{ and } y \not\lll x \\ \text{or } y <^\alpha x \text{ and } x \ll y. \end{cases}$$

Since the class of orderings $\preceq$ and the class of SPHs are equivalent, any way of revising one of these two types of structure will automatically give us a way of revising the other. We are free to use whichever one seems more appropriate at the time. For the purpose of expressing *desirable properties* of revising $\preceq$, it is easier to express such properties in terms of SPHs than $\preceq$.

4 Properties of SPH Revision

Given an SPH $\mathbb{S}$ and a sentence α, we want to determine the new SPH $\mathbb{S} \circledast \alpha$ which is the result of revising the entire SPH $\mathbb{S}$ by α. Assume $\mathbb{S} = (\lll, \ll, <)$ and let's denote $\mathbb{S} \circledast \alpha$ by $(\lll', \ll', <')$. Firstly, we have the following three fundamental properties:

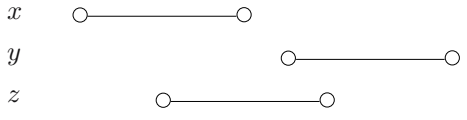
($\circledast$1) $\mathbb{S} \circledast \alpha$ is an SPH
($\circledast$2) $<' = <^*_\alpha$
($\circledast$3) If $\alpha \equiv \gamma$ then $\mathbb{S} \circledast \alpha = \mathbb{S} \circledast \gamma$

In ($\circledast$2), $<^*_\alpha$ is the strict version of the tpo $\leq^*_\alpha$ determined using $\leq$, $\ll$ and $\lll$ as in Proposition 1. In other words, $\mathbb{S} \circledast \alpha$ should be an SPH relative to $<^*_\alpha$. ($\circledast$3) is a syntax-irrelevance property.

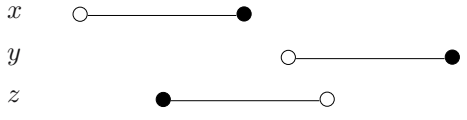
With $<'$ settled, it remains to specify $\lll'$ and $\ll'$. An initial suggestion for the new strong strict preferences $\lll'$ might be to keep it unchanged. That is, to set $\lll'$ equal to $\lll$. This can be seen as a pure application of minimal change to $\lll$. In addition, it is easy to see that $\lll \subseteq <'$ and so such a choice is not at odds with part *(ii)* of Theorem 1. However, the following example shows this can't be done in general. For $\mathbb{S} \circledast \alpha$ to be an SPH it is necessary to satisfy

(SPH1) $z \leq^*_\alpha x$ and $x \lll' y$ implies $z \lll' y$

But if we set $\lll = \lll'$ this might not hold in general. For suppose we are given a portion of the $\preceq$ corresponding to $\mathbb{S}$ as follows:



So $x \lll y$ and $z \not\lll y$. Now suppose we revise by a sentence α such that $z \in [\alpha]$ and $x, y \in [\neg\alpha]$.



Then $z <^*_\alpha x$, thus giving the required counterexample. Note, incidentally, that it is still a counterexample if we assume $y \in [\alpha]$. Thus there are times when the set of strong strict preferences *must* change. In the above counterexample, when we move from $\lll$ to $\lll'$ we must *either* lose $x \lll y$, *or* gain $z \lll y$. How do we decide which? A useful approach is to distinguish between the case $y \in [\neg\alpha]$, as indicated in the counterexample above, and the case $y \in [\alpha]$. In the former case intuition dictates that $x \lll y$ ought to be retained since α does not discriminate between x and y: they are both in $[\neg\alpha]$. Moreover, it is justifiable to gain $z \lll y$ since we have caught z on a good day ($z \in [\alpha]$) and y on a bad day ($y \in [\neg\alpha]$). On the other hand, in the case where $y \in [\alpha]$ it can be argued that the strong preference $x \lll y$ can be lost since we don't have such a strong case to prefer x over y anymore when $x \in [\neg\alpha]$ and $y \in [\alpha]$. Also, note that in this case it seems reasonable to require that the relative ordering of z and y with respect to $<$, $\ll$ and $\lll$ ought to remain unchanged since α does not distinguish between z and y: they are both in $[\alpha]$. This brings us to what can be regarded as the basic postulates for SPH revision, once ($\circledast$1)-($\circledast$3) are included as well:

($\circledast$4a) If $x \sim^\alpha y$ then $x \ll y$ iff $x \ll' y$
($\circledast$4b) If $x \sim^\alpha y$ then $x \lll y$ iff $x \lll' y$
($\circledast$5a) If $x <^\alpha y$ then $x \leq y$ implies $x \ll' y$
($\circledast$5b) If $x <^\alpha y$ then $x < y$ implies $x \lll' y$

Definition 2. *The SPH-revision operator* $\circledast$ *is* admissible *iff it satisfies (*$\circledast$*1)-(*$\circledast$*3), (*$\circledast$*4a), (*$\circledast$*4b), (*$\circledast$*5a) and (*$\circledast$*5b).*

We refer to this as admissible SPH revision since it corresponds closely to admissible revision as defined by [5]. ($\circledast$4a) and ($\circledast$4b) are versions of Darwiche and Pearl's (CR1) and (CR2) [6], or rules ($*3$) and ($*4$) defined earlier. They require that the ordering of two elements x and y be unchanged, wrt to $\ll$ and $\lll$, provided that the circumstances for x and y are the same (i.e. either both are in $[\alpha]$ or both are in $[\neg\alpha]$). This can be seen as an application of minimal change to $\ll$ and $\lll$. The postulates ($\circledast$5a) and ($\circledast$5b) are versions of rule ($*5$) defined earlier. In fact, in the presence of the fundamental rules ($\circledast$1) and ($\circledast$2), ($\circledast$5a) is a *strengthening* of ($*5$). They ensure that a "widening of the gap" between x and y occurs when x has a good day and y a bad day. This can be viewed as making sure that the evidence α is taken seriously. A world x in $[\alpha]$ will be more preferred with respect to a world y in $[\neg\alpha]$, provided that y was not preferred to x to start with. So, informally, admissible SPH revision effects a "slide to the right" of those worlds in $[\neg\alpha]$ in a manner similar to that described by Booth et al. [5]. The difference here is that, with the aid of $\ll$ and $\lll$, we can specify more precisely how such a slide is allowed to take place.

We now turn to some additional properties and investigate how they square up against admissible SPH revision. The first one we consider is

($\circledast$6) $\mathbb{S} \circledast \top = \mathbb{S}$

which states that everything remains unchanged if we revise by a tautology. And indeed, ($\circledast$6) follows immediately from ($\circledast$2), ($\circledast$4a) and ($\circledast$4b).

Next we consider the pair of properties

($\circledast$7a) If $x \ll y$ and $x \not\ll' y$ then $y <^{\alpha} x$
($\circledast$7b) If $x \lll y$ and $x \not\lll' y$ then $y <^{\alpha} x$

which state that losing a $\ll$-preference or a $\lll$-preference of x over y must be the result of y having a good day ($y \in [\alpha]$) and x a bad day ($x \in [\neg\alpha]$). It's easy to verify that ($\circledast$7a) follows from ($\circledast$4a) and ($\circledast$5a), while ($\circledast$7b) follows from ($\circledast$4b) and ($\circledast$5b).

Next is the pair of properties

($\circledast$8a) If $x \not\ll y$ and $x \ll' y$ then $x <^{\alpha} y$
($\circledast$8b) If $x \not\lll y$ and $x \lll' y$ then $x <^{\alpha} y$

which state that *gaining* a $\ll$-preference or an $\lll$-preference of x over y must be the result of x having a good day ($x \in [\alpha]$) and y a bad day ($y \in [\neg\alpha]$). It turns out that ($\circledast$8a) follows from ($\circledast$1), ($\circledast$2) and ($\circledast$4a), while ($\circledast$8b) follows from ($\circledast$1), ($\circledast$2) and ($\circledast$4b).

Next we mention a property *not* compatible with admissible SPH revision:

($\circledast$9) If $(\lll, \ll \cap <^*_\alpha, <^*_\alpha)$ is an SPH then $\mathbb{S} \circledast \alpha = (\lll, \ll \cap <^*_\alpha, <^*_\alpha)$

Property ($\circledast$9) is an attempt to enforce the principle of minimal change with respect to both $\ll$ and $\lll$. To see that it is incompatible with admissible revision, suppose $\mathbb{S}$ is of the form $(\emptyset, \emptyset, <)$, i.e., $\lll = \ll = \emptyset$. Assume furthermore that $x < y$ and suppose we then revise by α such that $x <^{\alpha} y$. Then $(\lll, \ll \cap <^*_\alpha, <^*_\alpha) =$

$(\emptyset, \emptyset, <^*_\alpha)$ is an SPH and so ($\circledast$9) dictates that $\mathbb{S} \circledast \alpha = (\emptyset, \emptyset, <^*_\alpha)$. But observe that admissible SPH revision, and more specifically ($\circledast$5b), requires that $x \lll' y$, which contradicts $\lll' = \emptyset$.

The difference between the approach advocated by ($\circledast$9) and admissible SPH revision is that ($\circledast$9) requires all three orderings to change as little as possible, while with ($\circledast$5a) and ($\circledast$5b) we are advocating that the new evidence α overrides the principle of minimal change.

Finally we mention a couple of plausible properties which go *beyond* those of admissible revision, in that they relate the results of revising by *different* sentences. We say sentences α, γ *agree on* worlds x, y iff either $[x <^\alpha y$ and $x <^\gamma y]$ or $[x \sim^\alpha y$ and $x \sim^\gamma y]$ or $[y <^\alpha x$ and $y <^\gamma x]$. That is, α and γ both "say the same thing" regarding the relative plausibility of x, y. The next 2 rules express that whether or not $x \ll' y$ and $x \lll' y$ should depend only on $\mathbb{S}$ and on what the input sentence says about the relative plausibility between x, y. They express a principle of "Independence of Irrelevant Alternatives in the Input". Here we are writing $\mathbb{S} \circledast \alpha = (\lll^*_\alpha, \ll^*_\alpha, <^*_\alpha)$ and $\mathbb{S} \circledast \gamma = (\lll^*_\gamma, \ll^*_\gamma, <^*_\gamma)$.

($\circledast$10a) If α and γ agree on x, y then $x \ll^*_\alpha y$ iff $x \ll^*_\gamma y$
($\circledast$10b) If α and γ agree on x, y then $x \lll^*_\alpha y$ iff $x \lll^*_\gamma y$

We omit the case for $<^*_\alpha$, $<^*_\gamma$, since it was already proved to follow from ($*1$)–($*7$) from the Section 2 [1]. It is thus already handled by ($\circledast$2). It can be shown that adding these two rules to those for admissible revision leads to the redundancy of ($\circledast$3) and allows ($\circledast$4a) and ($\circledast$4b) to be replaced by the simple rule ($\circledast$6).

5 A Concrete Revision Operator

In the previous section we proposed that any reasonable SPH-revision operator should at the very least be admissible according to Definition 2. In this section we demonstrate that such operators exist by defining a concrete admissible operator for SPH revision. This operator employs yet more structure which goes beyond SPHs and their corresponding orderings $\preceq$ over $W^\pm$, and which is a step closer to the PQI interval orders of Öztürk et al. [12] and also to semi-quantitative representations of epistemic states such as that of Spohn [8]. But we expect there will be other, interesting, admissible revision operators which can still be defined in a purely qualitative fashion. This is a topic for further research.

To decribe our operator it will be useful to switch back to the $\preceq$-representation of our tpo-revising structure rather than work directly with SPHs. The basic idea is to enrich the $\preceq$-representation with numerical information. More precisely we assume we are given upfront some fixed function p which assigns to each element $x^\epsilon \in W^\pm$ a real number $p(x^\epsilon)$ such that for all $x \in W$, $p(x^-) - p(x^+) = a > 0$, where a is some given real number which is also fixed upfront. The idea is that the smaller the number $p(x^\epsilon)$, the more preferred x^ϵ is. To each such assignment p we may associate an ordering $\preceq_p$ over $W^\pm$ given by $x^\epsilon \preceq_p y^\delta$ iff $p(x^\epsilon) \leq p(y^\delta)$. (But note that the mapping is not on-to-one – many different choices for p can yield the same ordering over $W^\pm$.) Essentially we replace our abstract intervals (x^+, x^-) with the real intervals $(p(x^+), p(x^-))$, all of length a. It is obvious that

$\preceq_p$ satisfies ($\preceq$1)–($\preceq$3). (Again, we point out it is not *absolutely* necessary for all the intervals to be of the *same* length a in order for $\preceq_p$ to satisfy ($\preceq$2).)

To revise a given SPH $\mathbb{S}$ by sentence α we will use the following procedure:

1. Convert $\mathbb{S}$ to its corresponding tpo $\preceq$ over $W^{\pm}$
2. Choose some p such that $\preceq = \preceq_p$
3. Revise p to get a new assignment $p * \alpha$
4. Take $\mathbb{S} \circledast \alpha$ to be the SPH corresponding to $\preceq_{p*\alpha}$

Clearly the crucial step here is step 3. How should we determine $p*\alpha$? We propose a very simple method here. We define $p * \alpha$ by setting, for each $x^{\epsilon} \in W^{\pm}$,

$$(p * \alpha)(x^{\epsilon}) = \begin{cases} p(x^{\epsilon}) & \text{if } x \in [\alpha] \\ p(x^{\epsilon}) + a & \text{if } x \in [\neg\alpha] \end{cases}$$

In other words, the interval $(p(x^+), p(x^-))$ associated to x remains unchanged if x satisfies α, but is "moved back" by amount a to $(p(x^-), p(x^-)+a)$ if x satisfies $\neg\alpha$. Essentially this boils down to nothing more than an operation familiar from the context of Spohn-type rankings known as *L-conditionalisation* [14].

The following result reveals what $\mathbb{S} \circledast \alpha$ will look like.

Proposition 2. *Assume* $\mathbb{S} = (\lll, \ll, <)$ *and let* $\mathbb{S} \circledast \alpha = (\lll', \ll', <')$ *be as defined in the above procedure, for suitable* p *in step 2. Then, for any* $x, y \in W$,
(i) $<' = <^*_\alpha$, *where* $*$ *is the revision operator corresponding to* $\mathbb{S}$ *as in Prop. 1.*
(ii)

$$x \ll' y \text{ iff } \begin{cases} x \sim^\alpha y \text{ and } x \ll y \\ \text{or } x <^\alpha y \text{ and } x \leq y \\ \text{or } y <^\alpha x \text{ and } p(x^-) + a \leq p(y^+). \end{cases}$$

(iii)

$$x \lll' y \text{ iff } \begin{cases} x \sim^\alpha y \text{ and } x \lll y \\ \text{or } x <^\alpha y \text{ and } x < y \\ \text{or } y <^\alpha x \text{ and } p(x^-) + a < p(y^+). \end{cases}$$

From this result we can see that $\circledast$ satisfies ($\circledast$2), ($\circledast$4a), ($\circledast$4b), ($\circledast$5a) and ($\circledast$5b). We can also see from this that the result of revision depends on $[\alpha]$ rather than α, thus ($\circledast$3) is also satisfied. Meanwhile rule ($\circledast$1) obviously holds. Thus:

Corollary 1. *The SPH-revision operator* $\circledast$ *defined via the above procedure from a given assignment* p *is admissible. Furthermore ($\circledast$10a) and ($\circledast$10b) also hold.*

6 Conclusion

Motivated by the problem of iterated revision of tpos, we extended the one-step revision framework of Booth et al. [1]. We revise not only the tpo, but also the *structure* required to guide the revision of the tpo. We showed that this structure may be described in terms of strict preference hierarchies (SPHs), and proved the equivalence of this representation with that already described by Booth et al.. We gave some properties which any reasonable SPH-revision operator should satisfy, and proved their consistency by giving a concrete example of an SPH-revision operator which satisfy them.

For future work we plan to investigate more desirable properties, and to examine useful equivalent ways to reformulate the ones we already have. In this paper all our properties are formulated as rules for single-step revision of SPHs. But since an SPH encodes the structure required to revise its associated tpo, these properties correspond to properties for *double-step* revision of tpos. To give an example, property (⊛5a) corresponds to the following rule governing revision of a tpo $\leq$ by α followed by β, which we denote for now by $\leq^*_{\alpha\cdot\beta}$:

$$\text{If } x <^{\alpha} y \text{ and } x \leq y \text{ then } x \leq^*_{\alpha\cdot\beta} y.$$

As mentioned above we intend to come up with other concrete SPH-revision operators, which perhaps can be described in purely qualitative terms rather than requiring extra numerical information like the operator described in this paper. Finally there seems to be a close connection between our work and the work done on preference modelling by Öztürk et al. [12]. The possible relationships between iterated belief revision and works such as these have, as far as we are aware, not been previously explored. We plan to look more closely at this.

References

1. Booth, R., Meyer, T., Wong, K.S.: A bad day surfing is better than a good day working: How to revise a total preorder. In: Proceedings of KR'06, pp. 230–238 (2006)
2. Hansson, S.O.: A Textbook of Belief Dynamics. Kluwer Academic Publishers, Dordrecht (1999)
3. Grove, A.: Two modellings for theory change. Journal of Philosophical Logic 17, 157–170 (1988)
4. Katsuno, H., Mendelzon, A.: Propositional knowledge base revision and minimal change. Artificial Intelligence 52, 263–294 (1991)
5. Booth, R., Meyer, T.: Admissible and restrained revision. Journal of Artificial Intelligence Research 26, 127–151 (2006)
6. Darwiche, A., Pearl, J.: On the logic of iterated belief revision. Artificial Intelligence 89, 1–29 (1997)
7. Nayak, A., Pagnucco, M., Peppas, P.: Dynamic belief change operators. Artificial Intelligence 146, 193–228 (2003)
8. Spohn, W.: Ordinal conditional functions: A dynamic theory of epistemic states. In: Causation in Decision: Belief, Change and Statistics, pp. 105–134. Kluwer Academic Publishers, Dordrecht (1988)
9. Papini, O.: Iterated revision operations stemming from the history of an agent's observations. In: Frontiers of belief revision, pp. 281–303. Kluwer Academic Publishers, Dordrecht (2001)
10. Hansson, S.O.: A survey of non-prioritized belief revision. Erkenntnis 50, 413–427 (1999)
11. Jin, Y., Thielscher, M.: Iterated belief revision, revised. In: Proceedings of IJCAI'05, pp. 478–483 (2005)
12. Öztürk, M., Tsoukiàs, A., Vincke, P.: Preference modelling. In: Multiple Criteria Decision Analysis: State of the Art Surveys, pp. 27–72. Springer, Heidelberg (2005)
13. Allen, J.: Maintaining knowledge about temporal intervals. Journal of the ACM 26, 832–843 (1983)
14. Goldszmidt, M., Pearl, J.: Qualitative probabilities for default reasoning, belief revision and causal modeling. Artificial Intelligence 84, 57–112 (1996)

Approaches to Constructing a Stratified Merged Knowledge Base

Anbu Yue[1], Weiru Liu[1], and Anthony Hunter[2]

[1] School of Electronics, Electrical Engineering and Computer Science,
Queen's University Belfast, Belfast BT7 1NN, UK
{a.yue, w.liu}@qub.ac.uk
[2] Department of Computer Science, University College London,
Gower Street, London WC1E 6BT, UK
a.hunter@cs.ucl.ac.uk

Abstract. Many merging operators have been proposed to merge either flat or stratified knowledge bases. The result of merging by such an operator is a flat base (or a set of models of the merged base) irrespective of whether the original ones are flat or stratified. The drawback of obtaining a flat merged base is that information about more preferred knowledge (formulae) versus less preferred knowledge is not explicitly represented, and this information can be very useful when deciding which formulae should be retained when there is a conflict. Therefore, it can be more desirable to return a stratified knowledge base as a merged result. A straightforward approach is to deploy the preference relation over possible worlds obtained after merging to reconstruct such a base. However, our study shows that such an approach can produce a poor result, that is, preference relations over possible worlds obtained after merging are not suitable for reconstructing a merged stratified base. Inspired by the Condorcet method in voting systems, we propose an alternative method to stratify a set of possible worlds given a set of stratified bases and take the stratification of possible worlds as the result of merging. Based on this, we provide a family of syntax-based methods and a family of model-based methods to construct a stratified merged knowledge base. In the syntax based methods, the formulae contained in the merged knowledge base are from the original individual knowledge bases. In contrast, in the model based methods, some additional formulae may be introduced into the merged knowledge base and no information in the original knowledge bases is lost. Since the merged result is a stratified knowledge base, the commonly agreed knowledge together with a preference relation over this knowledge can be extracted from the original knowledge bases.

1 Introduction

Preference (or priority) is very important in many fields of computer science, such as, in constraint satisfaction problems, in goal oriented decision making, and in system configurations. A preference relation can be used to represent an ordering over beliefs or goals. Preferences can be explicitly modelled in possibilistic logic (e.g. [5]) or using an ordinal conditional function (e.g., [9,11]), or a preference relation. With explicit preference information, a flat knowledge base can be extended to a stratified knowledge

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 54–65, 2007.

base, that is, propositions are divided into different strata according to the preferences (or priorities) they are given [2,3].

When multiple knowledge bases are available, one objective is to extract an overall view from them. This is known as knowledge base merging or belief merging. There are mainly two families of knowledge base merging operators: model-based ones which select some possible worlds (or interpretations) that are the *closest* to the original bases (e.g. [7,8,10]) and syntax-based ones which pick some formulae in the union of the original bases (e.g., [1,4]). It is worth noting that most of these merging operators (e.g. [7,8,1,4]) take flat knowledge bases as input, and only a few of them (e.g. [10]) allow the original knowledge bases to be stratified.

So far, all the merging operators, no matter whether model-based or syntax-based, return a flat knowledge base (or a set of models) as the merged result. We argue that, in practice, we may need a stratified knowledge base as the merged result which shows preference among formulae, irrespective of whether the original knowledge bases are flat or stratified. If a stratified knowledge base can be obtained as the result of merging, the preference over formulae can be very useful to resolve conflicts, that is, a more preferred formula should be retained in preference to a less preferred one if the two are in conflict.

Intuitively, it seems straightforward that a stratified merged base should be constructed easily from a model-based merging method using the preference relation over interpretations obtained after merging. However, our study of the merging methods in [8,10] shows that such a method is not adequate.

In [8], the commensurability assumption is required, and so a number[1] assigned to an interpretation (possible world) can be compared with another number assigned to another interpretation. On the other hand, although the commensurability assumption is not explicitly required in [10], each interpretation is still assigned a vector of numbers, each of which is the *priority* (or the absolute position) of the interpretation in the stratification of interpretations in relation to an individual knowledge base, and the numbers assigned in different stratifications are assumed comparable in order to establish the pre-order relation over vectors of numbers.

We argue that the numbers about distances or priorities obtained from different stratifications should not be comparable if we do not have the commensurability assumption, especially when knowledge bases are designed independently.

Inspired by the Condorcet method in *voting systems*, we provide an alternative method to define a preference relation, called *relative preference relation*, over interpretations. A relative preference relation considers whether an interpretation is more preferred than another collectively from a set of pre-order relations (or stratifications) over interpretations obtained from individual knowledge bases, but is independent of the absolute *priority* ([10]) or the *distance* ([8]) of an interpretation in relation to each single knowledge base, since these *numbers* are not assumed comparable. Then, the stratification of interpretations is constructed from this relative preference relation.

[1] This number can be the sum of distances between an interpretation and all the original knowledge bases or a vector of numbers each of which is the distance between the interpretation and a knowledge base.

Following this, we provide two families of methods, namely syntax-based and model-based, to construct a merged stratified knowledge base from the stratification of interpretations. The syntax-based methods assume that formulae in the stratified knowledge base are picked from the original bases. A disadvantage of these methods is that some implicit beliefs are lost. The model-based methods use models to reconstruct formulae in the merged base. These methods retain all the original knowledge and may also introduce additional formulae which do not appear in any original knowledge bases.

This paper is organized as follows. In Section 2, we introduce the preliminaries. In Section 3, we first explore a straightforward approach to constructing the stratification of interpretations after merging and to constructing a merged stratified knowledge base. We then provide an alternative approach to constructing the stratification of interpretations after merging by defining the concept of *relative preference relation*. In Section 4, we provide syntax-based methods for constructing a stratified knowledge base from a stratification of interpretations. Then, in Section 5, we provide model-based methods for constructing a stratified knowledge base. Finally, a brief comparison with related work and a short summary of the paper are given in Section 6.

2 Preliminary

2.1 Stratified Knowledge Base

We consider a propositional language $\mathcal{L}$ defined on a finite set $\mathcal{A}$ of propositional atoms, which are denoted by p, q, r etc. A proposition ϕ is constructed by propositional atoms with logic connections $\neg, \wedge, \vee, \rightarrow$ in the standard way. An interpretation ω (or possible world) is a function that maps $\mathcal{A}$ onto the set $\{0, 1\}$. The set of all possible interpretations on $\mathcal{A}$ is denoted as Ω. Function ω can be extended to any proposition in $\mathcal{L}$ in the usual way, $\omega : \mathcal{L} \rightarrow \{0, 1\}$. An interpretation ω is a model of (or satisfies) ϕ iff $\omega(\phi) = 1$, denoted as $\omega \models \phi$.

A *(flat) knowledge base* is a finite set of propositional formulae. A knowledge base K is consistent iff there is at least one interpretation that satisfies all propositions in K, and such interpretations are models of K. We use $Mod(K)$ to denote the set of models for K. $K \models \phi$ iff each model of K is a model of ϕ. For a set of models M, there exists a proposition ϕ_M s.t. $Mod(\phi_M) = M$. Theoretically, ϕ_M is non-deterministic since syntactically there can be more than one proposition that is satisfied by all the models in M.

A *stratified knowledge base* [3,2] is a finite set K of propositional formulae with a total pre-order relation $\preceq$ on K (a pre-order relation is a reflective and transitive relation, and $\preceq$ on K is total iff for all ϕ and ψ in K, either $\phi \preceq \psi$ or $\psi \preceq \phi$ holds). Intuitively, if $\phi \preceq \psi$ then ϕ is regarded as more certain, more preferred or more important than ψ. From the pre-order relation $\preceq$ on K, K can be stratified as $K = (S_1, \ldots, S_n)$, where S_i contains all the minimal propositions of set $\bigcup_{j=i}^{n} S_j$ w.r.t. $\preceq$, i.e., $S_i = \{\phi \in K \setminus (\cup_{j=1}^{i-1} S_j) : \forall \varphi \in K \setminus (\cup_{j=1}^{i-1} S_i), \phi \leq \varphi\}$. Each S_i is called a stratum of K and is non-empty. In the rest of this paper, we denote $\bigcup K = \bigcup_{i=1}^{n} S_i$. It is clear that for all ϕ and ψ in K, $\phi \preceq \psi$ iff $\phi \in S_i$, $\psi \in S_j$, and $i \leq j$. For simplicity, when we mention a knowledge base it is actually a stratified knowledge base unless it is stated otherwise in the rest of this paper.

2.2 Model Based Semantics

In [3,2], some model-based semantics are provided for stratified knowledge bases. In these methods, a pre-order relation on interpretations is induced from a knowledge base by an *ordering strategy*, and the minimal ones are regarded as the models of the knowledge base. Therefore, a non-classical consequence relation can be defined as $K \Vdash_X \phi$ iff $\omega(\phi) = 1$ for all ω such that ω is minimal w.r.t the pre-order relation $\preceq_X$ over Ω, where $\preceq_X$ is induced by K under the ordering strategy X. A strict relation $\prec_X$ is defined as $\omega \prec_X \omega'$ iff $\omega \preceq_X \omega'$ and $\omega' \not\preceq_X \omega$.

There are three widely used ordering strategies known as the *best out*, the *maxsat*, and the *leximin*. For a knowledge base $K = (S_1, \ldots, S_n)$, these ordering strategies are defined as follows.

- **best out ordering [2]** Let $r_{BO}(\omega) = \min_i\{\omega \not\models S_i\}$. Define $min_i\emptyset = +\infty$. $\omega \preceq_{bo} \omega'$ iff $r_{BO}(\omega) \geq r_{BO}(\omega')$.
- **maxsat ordering [3]** Let $r_{MO}(\omega) = \min_i\{\omega \models S_i\}$. $\omega \preceq_{maxsat} \omega'$ iff $r_{MO}(\omega) \leq r_{MO}(\omega')$.
- **leximin ordering [2]**: let $K_i(\omega) = \{\phi \in S_i \mid \omega \models \phi_i\}$. Then the leximin ordering $\preceq_{leximin}$ on Ω is defined as: $\omega \preceq_{leximin} \omega'$ iff
  - $|K_i(\omega)| = |K_i(\omega')|$ for all i, or
  - there is an i s.t. $|K_i(\omega)| > |K_i(\omega')|$, and $|K_j(\omega)| = |K_j(\omega')|$ for all $j < i$;

  where $|K_i|$ denotes the cardinality of set K_i.

Example 1. Let $K = (\{p\}, \{q\})$ be a knowledge base.

Table 1. Ranks calculated by different ordering strategies

ω	r_{BO}	r_{MO}	$\bar{r}_{Leximin}$
00	1	$+\infty$	$\langle 00 \rangle$
01	1	2	$\langle 01 \rangle$
10	2	1	$\langle 10 \rangle$
11	$+\infty$	1	$\langle 11 \rangle$

The i-th digit in vector $\bar{r}_{Leximin}(\omega)$ represents $|K_i(\omega)|$. For example, let $\omega = \{01\}$ represent possible world $\neg p q$, then $\bar{r}_{Leximin}(\omega) = \langle 01 \rangle$ means that $K_1(\omega) = 0$ and $K_2(\omega) = 1$. Then $\{11\}$ is the set of minimal models w.r.t the pre-order relations $\prec_{bo}$ and $\prec_{leximin}$, and set $\{11, 10\}$ contains all the minimal models w.r.t the pre-order relation $\prec_{maxsat}$. Therefore, $K \Vdash_{BO} p \wedge q$, $K \Vdash_{Maxsat} p$, and $K \Vdash_{Leximin} p \wedge q$.

From a pre-order relation $\preceq_X$ generated by the ordering strategy X from K, the interpretations in Ω can be stratified as $\Omega_{K,X} = (\Omega_1, \ldots, \Omega_m)$, where each Ω_i contains all the minimal interpretations from $\bigcup_{j=i}^{n} \Omega_j$ w.r.t. $\preceq_X$. Given $\Omega_{K_1} = (\Omega_1, \ldots, \Omega_n)$ and $\Omega_{K_2} = (\Omega'_1, \ldots, \Omega'_n)$, $\Omega_{K_1} = \Omega_{K_2}$ iff $\Omega_i = \Omega'_i$ for all i where $1 \leq i \leq n$. In the following, we may omit subscripts K, X from $\Omega_{K,X}$ when they are implicitly given.

As shown above, based on different ordering strategies, different conclusions are drawn from the same knowledge base K. Thus, selecting an ordering strategy for a given

knowledge base is important. Also, it is possible that the same stratification on interpretations can be induced from different knowledge bases under different ordering strategies.

3 Approaches to Stratifying the Set of Interpretations

In a model-based merging method, a pre-order relation on interpretations is constructed and a set of models (the minimal ones) is obtained as the result of merging flat or stratified knowledge bases. The resulting knowledge base is a flat base. Intuitively, it seems reasonable to recover a stratified merged knowledge base from the pre-order relation over interpretations. Following this idea, we take merging operators in [10,8] as examples and investigate if this approach is feasible. Our study below in Section 3.1 shows that a stratified merged knowledge base obtained this way can be counterintuitive. To overcome this problem, we propose an alternative method to stratify a set of interpretations using the concept of relative preference relation in Section 3.2.

3.1 A Simple Approach

In [10], a model-based merging method is proposed for merging stratified knowledge bases, however, the result is a flat knowledge base not a stratified one. The idea in the paper can be stated as follows. From each stratified knowledge base K with a chosen ordering strategy X, a stratification of interpretations is induced $\Omega_{K,X}$. In this way, an interpretation has a priority level w.r.t each K which is its priority level in $\Omega_{K,X}$. Then, each interpretation is associated with a vector of priority levels in relation to all the knowledge bases. Finally, a pre-order relation over interpretations is defined based on the lexicographical ordering over vectors of priorities and the interpretations which are minimal w.r.t this ordering relation are regarded as the models of the merged knowledge base. A straightforward approach to obtaining a stratified merged knowledge base is to construct strata directly from this pre-order relation over interpretations. Unfortunately, such a method is not as good as one may expect, as shown in the following example.

Example 2. Let $K_1 = (\{p\}, \{q\}, \{r\})$ *and* $K_2 = (\{r\}, \{q\}, \{p\})$ *be two knowledge bases. Using the leximin ordering strategy, two pre-order relations on interpretations can be induced from them respectively, and a pre-order relation for the merged knowledge base can be calculated under the leximin aggregation function as shown in Table 2. In Table 2, values in vector* $\bar{l}$ *are obtained by concatenating the numbers in the second and the third columns in ascending order.*

Based on the lexicographical ordering over vectors of priorities, a pre-order relation is defined on Ω *and it is stratified as* $\Omega = (\{111\}, \{110, 011\}, \{101\}, \{001, 100\}, \{010\}, \{000\})$. *The minimal model is* 111*, so, the merged flat knowledge base is equivalent to* $\{p \wedge q \wedge r\}$*, which seems reasonable. Note, in this paper we denote each model by a bit vector consisting of truth values of atoms e.g.* (p, q, r) *in this example. So* $\omega_1 = \{000\}$ *means* p, q *and* r *are all false.*

From this stratified Ω*, we can infer that* $p \wedge \neg q \wedge r$ *is less preferable than* $q \wedge (p \vee r) \wedge (\neg p \vee \neg r)$ *which means that, if only two of* p, q, r *are true, then* q *must be true. In other words,* q *is more certain than both* p *and* r*. On the other hand,* $\neg p \wedge q \wedge \neg r$ *is less preferable than* $\neg q \wedge (p \vee r) \wedge (\neg p \vee \neg r)$ *which implies that when only one of* p, q, r

Table 2. Constructing the stratification of interpretations

ω	$\Omega_{K_1,leximin}$	$\Omega_{K_2,leximin}$	$\bar{l}$
000	8	8	$\langle 88 \rangle$
001	7	4	$\langle 47 \rangle$
010	6	6	$\langle 66 \rangle$
011	5	2	$\langle 25 \rangle$
100	4	7	$\langle 47 \rangle$
101	3	3	$\langle 33 \rangle$
110	2	5	$\langle 25 \rangle$
111	1	1	$\langle 11 \rangle$

holds, it should not be q, that is, q is less certain than both p and r now. This contradicts with the previous inference. Therefore, taking the stratification of interpretations as a way to construct a merged knowledge base may imply counterintuitive results.

If we take a flat knowledge base as a special case of a stratified knowledge base with only one stratum, the merging methods in DA^2 family [8] can be viewed as special cases of merging stratified knowledge bases. Similarly, in these methods, a set of models for a flat knowledge base is given as the result of an operator.

Example 3. Let $K_1 = K_2 = K_3 = K_4 = K_5 = \{p\}$ and $K_6 = K_7 = K_8 = K_9 = \{\neg p\}$ be nine knowledge bases. Five of them say that p is true and four say that p is false. Merging by δ^{d_D,sum_1,sum_2}, a specific operator in DA^2 [8], the model with p is true (having true value 1) is the only model for the merged knowledge base. In this merging operator, d_D is a distance measure between a formula and a possible world and it is defined as $d_D(\omega, \phi) = 0$ if $\omega \models \phi$, otherwise $d_D(\omega, \phi) = 1$. $\mathsf{sum}_1(\omega, K_i) = \Sigma_{\phi \in K_i} d_D(\omega, \phi)$ and $\mathsf{sum}_2(\omega, P) = \Sigma_{K_i \in P} \mathsf{sum}_1(\omega, K_i)$, which is the sum of distances between knowledge bases K_i in a knowledge profile P (a knowledge profile is a finite set of knowledge bases) and a possible world. In Table 3, values in the second to the tenth columns are the distances between an interpretation and a knowledge base (using sum_1), values in the eleventh column are from sum_2.

Now if we revise $K_6, ..., K_9$ as $K'_6 = K'_7 = \{\neg p, \neg p \wedge q\}$ and $K'_8 = K'_9 = \{\neg p, \neg p \wedge \neg q\}$ respectively, then semantically, these four knowledge bases together state the same conclusion as K_6, ..., K_9 do. That is, both sets of knowledge bases say p is false. However, when we replace $K_6, ..., K_9$ with $K'_6, ..., K'_9$ and merge them with $K_1, ..., K_5$, we get a different merged knowledge base K', whose models are $\{00, 01\}$ as shown in Table 4. Obviously, $K' \models \neg p$, $K' \not\models q$ and $K' \not\models \neg q$.

Table 3. Merging knowledge bases using an operator in DA^2

ω	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	sum_2
$\neg p$	1	1	1	1	1	0	0	0	0	5
p	0	0	0	0	0	1	1	1	1	4

Table 4. Merging knowledge bases with an operator in DA^2

ω	K_1	K_2	K_3	K_4	K_5	K'_6	K'_7	K'_8	K'_9	sum
00	1	1	1	1	1	1	1	0	0	7
01	1	1	1	1	1	0	0	1	1	7
10	0	0	0	0	0	2	2	2	2	8
11	0	0	0	0	0	2	2	2	2	8

The reason is that, in DA^2, the commensurability assumption is required. Under this assumption, although $K'_6, ..., K'_9$ collectively draw the same conclusion as $K_6, ..., K_9$, when they are merged with the other bases, the preferability of the statement p is implicitly decreased. That is why the merged result is changed.

To summarize, we believe that using the information on stratification over interpretations from a merging operator to construct a stratified knowledge base can not return a satisfactory result.

3.2 Stratifying Interpretations by Relative Preference Relation

In the methods discussed above, an interpretation is associated with a number (or a vector of numbers) about its priority level(s) that determines which stratum (or strata) it is in and this number (or a vector of numbers) is the absolute position(s) (stratum/strata) it reflects. We argue that, the absolute position of an interpretation in a stratification is not so important, since one could not tell how the preferences among items of beliefs in other knowledge bases are determined. For instance, when one knowledge base regards that $Mod(p)$ are more preferable than $Mod(q)$ and $Mod(q)$ are more preferable than $Mod(r)$, then $Mod(r) \setminus (Mod(p) \cup Mod(q))$ are the third level of models to be preferred. If other knowledge bases do not consider q, then these models (which are for r) are underestimated if a merging process considers only the absolute position that a model occurs in each stratification of interpretations.

We believe that only the relative preferences between interpretations induced from a knowledge base by ordering strategy is meaningful in a merging process:

Definition 1 (Relative Preference Relation). *Let $\{\Omega_{K_1,X_1}, \ldots, \Omega_{K_n,X_n}\}$ be a multi-set. We define a binary relative preference relation $R : \Omega \times \Omega$ as:*
$\mathsf{R}(\omega, \omega')$ *iff* $|\{\Omega_{K_i,X_i} s.t. \omega \prec_i \omega'\}| < |\{\Omega_{K_i,X_i} s.t. \omega' \prec_i \omega\}|$ *where $\prec_i$ is the strict partial order induced from Ω_{K_i,X_i}.*

$\mathsf{R}(\omega, \omega')$ means that more knowledge bases prefer ω than ω'. A relative preference relation is partial, anti-symmetric and irreflective, and it is not transitive, so it is not a total pre-order relation.

Definition 2 (Undominated Set). *Let* R *be a relative preference relation over $\Omega \times \Omega$ and let Q be a subset of Ω. Q is called an undominated set of Ω, if*

$$\forall \omega \in Q, \forall \omega' \in \Omega \setminus Q, \mathsf{R}(\omega', \omega) \text{ does not hold}$$

Q is a minimal undominated set *of Ω if for any undominated set P of Ω, $P \subset Q$ does not hold.*

We denote the set of minimal undominated sets of Ω w.r.t. R as U^R_Ω.

Definition 3. *Let* R *be a relative preference relation. A stratification of interpretations $\Omega = (\Omega_1, \ldots, \Omega_n)$ can be obtained from* R *such that $\Omega_i = \cup Q$ where $Q \in U^R_{\Omega \setminus \cup_{j=1}^{i-1} \Omega_j}$ based on Definition 2.*

This way, the stratification of interpretations is independent of absolute priorities (or positions) of interpretations and the commensurability assumptions is not required.

Since from a stratification of interpretations, a total pre-order relation can be induced, the above definition also defines a total pre-order relation over interpretations.

Example 4. Let $K_1 = (\{p\}, \{q\}, \{r\})$ and $K_2 = (\{r\}, \{q\}, \{p\})$ be two knowledge bases. If we apply ordering strategy leximin on both bases, we get two stratifications on Ω as

$\Omega_{K_1, leximin} = (\{111\}, \{110\}, \{101\}, \{100\}, \{011\}, \{010\}, \{001\}, \{000\})$

$\Omega_{K_2, leximin} = (\{111\}, \{011\}, \{101\}, \{001\}, \{110\}, \{010\}, \{100\}, \{000\})$

Then a relative preference relation over Ω can be defined based on them. From this relative preference relation, we can get a final stratification on Ω as

$\Omega = (\{111\}, \{110, 011, 101\}, \{001, 010, 100\}, \{000\})$.

Obviously, p, q, r are symmetric and thus are equally preferred and this stratification is better than that obtained in Example 2.

4 Syntax-Based Approaches to Constructing Stratified Knowledge Bases

Based on the stratification of interpretations obtained in the above section, we explore approaches to stratifying a merged knowledge base. We discuss syntax-based methods in this section and investigate model-based methods in the next section.

In the syntax-based methods, we assume that we *pick* some (may not be all) propositions from the original knowledge bases and stratify them based on a stratification of interpretations.

Definition 4. *Let $\Omega = (\Omega_1, ..., \Omega_n)$ be a stratification of interpretations and S be a set of propositions. Let X be an ordering strategy. A stratified knowledge base $K_S^{X,\Omega} = (S_1, ..., S_m)$ is an X* **dominated construction** *from S w.r.t. Ω if $\bigcup_{i=1}^m S_i \subseteq S$ and $\Omega_{K_S^{X,\Omega}, X} = \Omega$.*

Definition 5 (best out construction). *Let $\Omega = (\Omega_1, \ldots, \Omega_n)$ be a stratification of interpretations and S be a set of propositions. We define $K_S^{bo,\Omega} = (S_1, \ldots, S_{n-1})$ where $S_i = \{\phi \in S \mid \forall \omega \in \Omega_j, \omega \models \phi, \forall\, j \in [1, n-i]\} \setminus \bigcup_{j=1}^{i-1} S_j$ and $S_i \neq \emptyset$.*

Proposition 1. *Let Ω be a stratification of interpretations and S be a set of propositions. If there exists a stratified knowledge base K s.t. $\Omega_{K,bo} = \Omega$ and $\bigcup K \subseteq S$, then $K_S^{bo,\Omega}$ defined in Definition 5 is a best out dominated construction from S w.r.t. Ω, that is $\bigcup_{i=1}^{n-1} S_i \subseteq S$ and $\Omega_{K_S^{bo,\Omega}, bo} = \Omega$.*

Example 5. Let $\Omega = (\{011\}, \{111\}, \{101\}, \{000, 010, 100, 110, 001\})$ *and let the set of propositions be* $S = \{p \vee q, r, q \vee \neg r, \neg p \vee \neg r, \neg p \wedge \neg q\}$*, then we can get a stratified knowledge base based on* S *as* $K_S^{bo,\Omega} = (\{p \vee q, r\}, \{q \vee \neg r\}, \{\neg p \vee \neg r\})$ *which satisfies* $\Omega_{K_S^{bo,\Omega},bo} = \Omega$*. This implies that there is a stratified knowledge base* K *such that* $\bigcup K \subseteq S$ *and* $\Omega_{K,bo} = \Omega$*.*

However, if we have $S' = \{p \vee q, r, \neg q \vee \neg r, \neg p \vee \neg r, \neg p \wedge \neg q, \neg p \vee q\}$*, then we have* $K_{S'}^{bo,\Omega} = (\{p \vee q, r\}, \{\neg p \vee q\}, \{\neg p \vee \neg r\})$*. In this case,* $\Omega_{K_{S'}^{bo,\Omega},bo} = (\{011\}, \{111\}, \{001, 101\}, \{000, 010, 100, 110\}) \neq \Omega$*, which means that* $\nexists K$ *such that* $\Omega_{K,bo} = \Omega$ *and* $\bigcup K \subseteq S'$*.*

Definition 6 (maxsat construction). *Let* $\Omega = (\Omega_1, \ldots, \Omega_n)$ *be a stratification of interpretations and* S *be a set of propositions. We define* $K_S^{maxsat,\Omega} = (S_1, \ldots, S_n)$ *where* $S_i = \{\phi \in S \mid \forall \omega \in \Omega_i, \omega \models \phi\} \setminus \bigcup_{j=1}^{i-1} S_j$ *and* $S_i \neq \emptyset$*.*

Proposition 2. *Let* Ω *be a stratification of interpretations and* S *be a set of propositions. If there exists a stratified knowledge base* K *s.t.* $\Omega_{K,maxsat} = \Omega$ *and* $\bigcup K \subseteq S$*, then* $K_S^{maxsat,\Omega}$ *is a maxsat-dominated construction from* S *w.r.t.* Ω*, that is* $\bigcup_{i=1}^{n-1} S_i \subseteq S$ *and* $\Omega_{K_S^{maxsat,\Omega},maxsat} = \Omega$*.*

Definition 7 (leximin construction). *Let* $\Omega = (\Omega_1, \ldots, \Omega_n)$ *be a stratification of interpretations and* S *be a set of propositions. We define* $K_S^{leximin,\Omega} = (S_1, \ldots, S_n)$ *where* $S_i = \{\phi \in S \mid \forall \omega \in \Omega_i, \omega \models \phi$*, where* $\forall j > i, \forall \omega \in \Omega_j, \omega \not\models \phi\}$ *and* $S_i \neq \emptyset$*.*

Proposition 3. *Let* Ω *be a stratification of interpretations and* S *be a set of propositions. If there exists a stratified knowledge base* $K = (S_1, \ldots, S_n)$ *s.t. each* S_i *is a singleton set,* $\Omega_{K,leximin} = \Omega$ *and* $\bigcup K = S$*, then* $K_S^{leximin,\Omega}$ *is a leximin dominated construction from* Ω*, i.e.* $\Omega_{K_S^{leximin,\Omega},leximin} = \Omega$*.*

In this proposition, it is required that $\bigcup K = S$, because the leximin ordering strategy is more syntax sensitive than best out and maxsat.

Example 6. Let $\Omega = (\{011\}, \{101, 111\}, \{000, 010, 100, 110\}, \{001\})$ *and let* $S = \{(p \vee q) \wedge r, q \vee \neg r, \neg p \vee \neg r\}$*. Then we have a stratified knowledge base* $K_S^{leximin,\Omega} = (\{(p \vee q) \wedge r\}, \{q \vee \neg r\}, \{\neg p \vee \neg r\})$ *which satisfies* $\Omega_{K_S^{leximin,\Omega},leximin} = \Omega$*. This implies that* $\exists K$ *such that* $\Omega_{K,leximin} = \Omega$ *and* $\bigcup K = S$*.*

For the above three methods, it is assumed that we know what propositions should appear in the merged stratified knowledge base. This assumption comes from the intuition that when merging stratified knowledge bases, only those propositions that appear in some knowledge bases would be considered. This is consistent with the ideas in syntax-based merging operators. However when merging knowledge bases, some implicit knowledge can be drawn and such knowledge does not necessarily appear in any of the individual knowledge bases.

Example 7. Let $K_1 = (\{p \wedge q\})$ *and* $K_2 = (\{\neg p \wedge q\})$*. From this, we can get two stratifications on* Ω *using ordering strategy leximin as*

$$\Omega_{K_1,leximin} = (\{11\}, \{00, 01, 10\})$$
$$\Omega_{K_2,leximin} = (\{01\}, \{00, 10, 11\})$$

Based on these two stratifications, it is possible to define a relative preference relation R, and then Ω can be stratified as

$$\Omega_K = (\{01, 11\}, \{00, 10\})$$

Through this stratification, we can infer that q should be true and p be unknown (or undefined) in the merged base, if we take the models in the first stratum as the models of merging. But q as a proposition does not appear in K_1 or K_2. If we attempt to reconstruct a stratified merged knowledge base from the set $S = K_1 \cup K_2 = \{p \wedge q, \neg p \wedge q\}$ directly with either best out, or maxsat, or leximin, we can only get $K_S^{\Omega_K, X} = (\emptyset)$.

So, if we restrict S to be as $S \subseteq \bigcup_i (\cup K_i)$ then implicit knowledge will be lost. One way to overcome this is to allow S to be a bigger set, such as (a trivial one) S could be $\bigcup Cn(\bigcup K)$, where Cn is the classical deductive closure operator.

5 Model Based Approaches to Constructing Stratified Knowledge Bases

An alternative to the syntax-based family of methods is to construct propositions for the merged knowledge base directly from the stratification of interpretations $\Omega = (\Omega_1, \ldots, \Omega_n)$, rather than *picking* propositions from the original knowledge bases. In this section, we investigate how such an approach can be established.

Definition 8. *Let $\Omega = (\Omega_1, \ldots, \Omega_n)$ be stratification of interpretations. Define $K^{bo,\Omega} = (S_1, \ldots, S_{n-1})$, where $S_i = \phi_{(\bigcup_{j=1}^{n-i} \Omega_j)}, i = 1, \ldots, n-1$.*

Proposition 4. *Let Ω be a stratification of interpretations. Then $\Omega_{K^{bo,\Omega}, bo} = \Omega$.*

Example 8. Let $\Omega = (\{11\}, \{10\}, \{00, 01\})$. Then $K^{bo,\Omega} = (\{p\}, \{p \wedge q\})$, and $\Omega_{K^{bo,\Omega}, bo} = \Omega$.

Definition 9. *Let $\Omega = (\Omega_1, \ldots, \Omega_n)$ be a stratification of interpretations. Define $K^{maxsat,\Omega} = (S_1, \ldots, S_{n-1})$, where $S_i = \phi_{(\bigcup_{j=1}^{i} \Omega_j)}$, $i = 1, \ldots, n-1$.*

Proposition 5. *Let Ω be a stratification of interpretations. Then $\Omega_{K^{maxsat,\Omega}, maxsat} = \Omega$.*

Example 9. Let $\Omega = (\{11\}, \{10\}, \{00, 01\})$. Then $K^{maxsat,\Omega} = (\{p \wedge q\}, \{p\})$, and $\Omega_{K^{maxsat,\Omega}, maxsat} = \Omega$.

Definition 10. *Let $\Omega = (\Omega_1, \ldots, \Omega_n)$ be a stratification of interpretations. Define $\Omega' = (\Omega'_0, \ldots, \Omega'_{2^l - 1})$ as:*

1. *$\Omega'_i = \emptyset$, $i \in [0, 2^l - n - 1]$*
2. *$\Omega'_{2^l - n + i - 1} = \Omega_i$, $i \in [1, n]$*

where l is the smallest number s.t. $2^l \geq n$.

Let $S_i = \phi_{\bigcup_{\pi(j,i)=0} \Omega'_j} (1 \leq i \leq l)$, where $\pi(j, i) = 0$ if $(j \mod 2^{l-i+1}) < 2^{l-i}$, otherwise $\pi(j, i) = 1$.

Then we define $K^{Leximin,\Omega} = (S_1, \ldots, S_l)$.

In this definition, $\pi(j,i)$ is in fact the value of i^{th} (from the left hand) digit of j when j is represented as a binary value with l-bits. For example, if we set $l=3$ and we have $j=3$, then j can be represented as a binary value 011, so $\pi(j,2)=1$, since the second digit of 011 is 1. We also have $(3 \mod 2^{3-2+1})=3$ and $3 \geq 2^{3-2}$, so $\pi(3,2)=1$ too.

Proposition 6. *Let Ω be a stratification of interpretations. Then $\Omega_{K^{leximin,\Omega},leximin} = \Omega$.*

Example 10. Let $\Omega = (\{11\},\{10\},\{00,01\})$. Then $K^{leximin,\Omega} = (\{p \wedge q\},\{p \wedge \neg q\})$, and $\Omega_{K^{leximin,\Omega},leximin} = \Omega$.

When the interpretations are stratified into relatively a large number of strata, the $leximin$ dominated construction method can drastically reduce the number of strata of the merged knowledge base compared to both best out and maxsat.

Example 11. Let $\Omega = (\{111\},\{110\},\{101\},\{100\},\{011\},\{010\},\{001\},\{000\})$. Then

$$K^{leximin,\Omega} = (\{p\},\{q\},\{r\})$$

However, the other two strategies both return a knowledge base with a lot more propositions. That is

$$K^{bo,\Omega} = (\{p \vee q \vee r\},\{p \vee q\},\{(p \vee q) \wedge (p \vee r)\},\{p\},\{p \wedge (q \vee r)\},\{p \wedge q\},\{p \wedge q \wedge r\})$$

and

$$K^{maxsat,\Omega} = (\{p \wedge q \wedge r\},\{p \wedge q\},\{p \wedge (q \vee r)\},\{p\},\{(p \vee q) \wedge (p \vee r)\},\{p \vee q\},\{p \vee q \vee r\})$$

6 Conclusion

In knowledge base merging, most existing methods merge either flat or stratified knowledge bases and produce a flat base (or a set of models) as the result (e.g., [6,7,8,10]). We argue that ideally a stratified merged base would be better since it has additional information about which formulae are more preferred than others. Motivated by this, we investigated how such a stratified merged base can be constructed.

We first looked at the possibility of recovering a stratified base based on the stratification of interpretations obtained after applying a merging operator. However, the results show that such a straightforward method can produce counterintuitive results. The main reason is that almost all the merging methods, especially the model-based ones (which return a set of models as the merged results), require an assumption of commensurability, so the absolute position of each interpretation in each stratification of interpretations is important. We argue that only the relative position of an interpretation w.r.t other interpretations is important if we do not require this commensurability. Based on this, we proposed a method to define a binary relative preference relation between interpretations and then this relation is used to stratify interpretations given a set of individual stratifications of interpretations induced from the original knowledge bases.

Following this, we proposed a family of syntax-based and model-based approaches to stratifying a merged knowledge base. Properties of these stratification approaches are also studied.

The idea of constructing a relative preference relation is inspired by Condorcet methods in *voting systems* or the *social choice theory*. The winner of votes by the Schulze method, an instance of Condorcet methods, is exactly the same as the most preferred interpretations in our approach to generate the stratification of interpretations using the relative preference relation, when we treat a candidate as an interpretation and a boll as a stratification of interpretations.

For future work, we will further investigate appropriate approaches to stratifying merged knowledge bases and to discuss additional logical properties of our methods. In contrast with merging flat knowledge bases, we believe that merging stratified knowledge bases should put more emphasize on considering preferences of propositions and thus should satisfy a different set of constraints or postulates to those for merging flat bases.

Acknowledgement. This work is funded by the EPSRC projects with reference numbers: EP/D070864/1 and EP/D074282/1.

References

1. Baral, C., Kraus, S., Minker, J.: Combining multiple knowledge bases. IEEE Transactions on Knowledge Data Engineering 3, 208–220 (1991)
2. Benferhat, S., Cayrol, C., Dubois, D., Lang, J., Prade, H.: Inconsistency management and prioritized syntax-based entailment. In: Proc. Of IJCAI'93, pp. 640–647 (1993)
3. Brewka, G.: A rank based description language for qualitative preferences. In: The 16th European Conference on Artificial Intelligence (ECAI'04), pp. 303–307 (2004)
4. Delgrande, J.P., Schaub, T.: Consistency-based approaches to merging knowledge bases: preliminary report. In: The 10th International Workshop on Non-Monotonic Reasoning (NMR2004), pp. 26–133 (2004)
5. Dubois, D., Lang, J., Prade, H.: Possibilistic logic. In: Gabbay, D., Hogger, C.J., Robinson, J.A. (eds.) Handbook of Logic in Artificial Intelligence and Logic Programming, Nomonotonic Reasoning and Uncertain Reasoning, vol. 3, pp. 439–513. Oxford University Press, Oxford (1994)
6. Everaere, P., Konieczny, S., Marquis, P.: On merging strategy-proofness. In: The 9th International Conference on Principles of Knowledge Representation and Reasoning (KR'04), pp. 357–368 (2004)
7. Konieczny, S., Pérez, R.P.: On the logic of merging. In: Proc. Of KR'98, pp. 488–498 (1998)
8. Konieczny, S., Lang, J., Marquis, P.: DA^2 merging operators. Artificial Intelligence 157(1-2), 49–79 (2004)
9. Meyer, T.: On the semantics of combination operations. Journal of Applied Non-Classical Logics 11(1-2), 59–84 (2001)
10. Qi, G., Liu, W., Bell, D.A.: Merging stratified knowledge bases under constraints. In: The 21st American National Conference on Artificial Intelligence (AAAI'06), pp. 281–286 (2006)
11. Spohn, W.: Ordinal conditional functions. a dynamic theory of epistemic states. In: Harper, W., Skyrms, B. (eds.) Causation in Decision, Belief Change, and Statistics, vol. 2, pp. 105–134. Kluwer, Dordrecht (1988)

Syntactic Propositional Belief Bases Fusion with Removed Sets

Julien Hue[1], Odile Papini[2], and Eric Wurbel[1]

[1] LSIS-CNRS 6168, Université du Sud Toulon -Var. BP 132. 83957 La Garde Cedex France
{hue, wurbel}@univ-tln.fr
[2] LSIS-CNRS 6168, ESIL, Université de la Méditerranée. Av de Luminy. Case 13009 Marseille Cedex 9 France
papini@esil.univmed.fr

Abstract. The problem of merging multiple sources information is central in several domains of computer science. In knowledge representation for artificial intelligence, several approaches have been proposed for propositional bases fusion, however, most of them are defined at a semantic level and are untractable. This paper proposes a new syntactic approach of belief bases fusion, called Removed Sets Fusion (RSF). The notion of removed-set, initially defined in the context of belief revision is extended to fusion and most of the classical fusion operations are syntactically captured by RSF. In order to efficiently implement RSF, the paper shows how RSF can be encoded into a logic program with answer set semantics, then presents an adaptation of the smodels system devoted to efficiently compute the removed sets in order to perform RSF. Finally a preliminary experimental study shows that the answer set programming approach seems promising for performing belief bases fusion on real scale applications.

1 Introduction

Merging information coming from different sources is an important issue in various domains of computer science like knowledge representation for artificial intelligence, decision making or databases. The aim of fusion is to obtain a global point of view, exploiting the complementarity between sources, solving different existing conflicts, reducing the possible redundancies. Among the various approaches of multiple sources information merging, logical approaches gave rise to increasing interest in the last decade [1,2,3,4,5]. Most of these approaches have been defined within the framework of classical logic, more often propositional, and have been semantically defined. Different postulates characterizing the rational behavior of fusion operators have been proposed [6] and various operators have been defined according to whether explicit or implicit priorities are available [6], [7], [8], [9], [10], [11]. More recently, new approaches have been proposed like semantic merging for propositional bases, stemming from the Hamming distance [12] or syntactic fusion in a possibilistic framework [13,14] which is a real advantage at a computational point of view.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 66–77, 2007.

This paper proposes a new approach for performing syntactic fusion of propositional belief bases. We show that the classical fusion operations $Card$, Σ, Max, $GMax$, initially defined at the semantic level, can be expressed within our syntactic framework. We then show that an efficient implementation of these operations, based on answer set programming, can be performed. In particular this paper focuses on the following three issues:

- We extend the Removed Sets Revision to the fusion of propositional belief bases, called Removed Sets Fusion (RSF). We show how the notion of removed-set, roughly speaking, the subsets of clauses to remove to restore consistency, initially defined in the context of belief bases revision [15,16] is generalized to the case of belief bases fusion. We then show that classical fusion operations are captured within this framework since each fusion strategy is encoded by a preference relation between subsets of clauses.
- In the last decade, answer set programming has been considered as a convenient tool to handle non-monotonic reasoning systems. Moreover, several efficient systems have been developed [17], [18], [19], [20], [21]. We propose to formalize the Removed Sets Fusion in terms of answer set programming and to adapt the smodels system in order to compute preferred answer sets which correspond to removed sets. Therefore, to propose an effective computational fusion algorithm.
- The conducted preliminary experimental study illustrates the behaviour of RSF for the $Card$, Σ strategies and seems promising for performing fusion in real scale applications.

The rest of this paper is organized as follows. The next section fixes the notations and gives a refresher on fusion, removed set revision and on answer set programming. The paper then presents the Removed Set Fusion. It shows how Removed Sets Fusion is encoded into logic programming with answer set semantics and presents an adaptation of the Smodels system for computing answer sets and performing Removed Sets Fusion. It then presents a preliminary experimental study which illustrates the approach and shows that the answer set programming implementation seems promising before concluding.

2 Background and Notations

We consider a propositional language $\mathcal{L}$ over a finite alphabet $\mathcal{P}$ of atoms. A literal is an atom or the negation of an atom. The usual propositional connectives are denoted by $\neg$, $\wedge$, $\vee$ and Cn denotes the logical consequence. A *belief base* K is a finite set of propositional formulae over a propositional language $\mathcal{L}$.

2.1 Fusion

Let $E = \{K_1, \ldots, K_n\}$ be a multi-set of n consistent belief bases to be merged, E is called a *belief profile*. The n belief bases $K_1, \ldots, K_n$ are not necessarily different and the union of belief bases, taking repetitions into account, is denoted by $\sqcup$

and their conjunction and disjunction are denoted by $\bigwedge$ and $\bigvee$ respectively. For the sake of simplicity, we denote by K the belief set consisting of the singleton $E = \{K\}$.

We define a fusion operator Δ as a function which associates to each belief profile a classical consistent belief base denoted by $\Delta(E)$. In the literature, there are two different ways for defining $\Delta(E)$: either using some implicit priority or not. In the following implicit priority is not assumed.

There are two straighforward ways for defining $\Delta(E)$ depending if the sources are conflicting or not, the classical conjunctive merging : $\Delta(E) = \bigwedge_{K_i \in E} K_i$ suitable when the sources are not conflicting and the classical disjunctive merging : $\Delta(E) = \bigvee_{K_i \in E} K_i$ appropriate in case of conflicting sources. These two opposite cases are not satisfactory, then several methods have been proposed for fusion according to whether the bases have the same importance or not.

In particular, the following classical fusion operators have been proposed. The *Cardinality operator*, denoted by $Card$, [1] which takes the number of the belief bases of E into account. The *Sum operator*, denoted by Σ, [22,2]which follows the point of view of the majority of the belief bases of E.

The *Max-based* operator, denoted by Max [4], which tries to satisfy all the belief bases of E. The *Leximax-based* operator, denoted by $GMax$, [6] which tries to satisfy all the belief bases of E, taking the belief bases into account, according to a lexicographic ordering over them.

Different postulates characterizing the rational behaviour of fusion operators have been proposed [6]. Moreover, the various operators have been classified according to two families: the majority and the arbitration ones.

2.2 Answer Sets

A *normal logic program* is a set of rules of the form $c \leftarrow a_1, ..., a_n, not\ b_1, ..., not\ b_m$ where $c, a_i (1 \leq i \leq n), b_j (1 \leq j \leq m)$ are propositional atoms and the symbol *not* stands for *negation as failure*. For a rule r like above, we introduce $head(r) = c$ and $body(r) = \{a_1, ..., a_n,\ b_1, ..., b_m\ \}$. Furthermore, let $body^+(r) = \{a_1, ..., a_n\}$ denotes the set of positive body atoms and $body^-(r) = \{b_1, ..., b_m\}$ the set of negative body atoms, and $body(r) = body^+(r) \cup body^-(r)$.

Let r be a rule, r^+ denotes the rule $head(r) \leftarrow body^+(r)$, obtained from r by deleting all negative body atoms in the body of r.

A set of atoms X is *closed under* a basic program P iff for any rule $r \in P$, $head(r) \in X$ whenever $body(r) \subseteq X$. The smallest set of atoms which is closed under a basic program P is denoted by $CN(P)$.

The *reduct* or Gelfond-Lifschitz transformation [23], P^X of a program P relatively to a set X of atoms is defined by $P^X = \{r^+ \mid r \in P \text{ and } body^-(r) \cap X = \emptyset\}$.

A set of atoms X is an *answer set* of P iff $CN(P^X) = X$.

Definition 1. *Let L be a set of literals and A be a set of atoms. L covers A iff $A \subseteq Atom(L)$.*

2.3 Smodels

Smodels is the first and simplest answer set solver [24]. It's a Branch and Bound algorithm (see Algorithm 1) that builds, as one goes along, a set of atoms A representing a potential answer set. It uses the following functions: $expand(A)$ which computes the immediate consequences of A, $conflict(A)$ which detects the conflicts that may arise after the expansion and $heuristic(A)$ which tries to reduce the search space by maximizing the number of deduced atoms. The function $heuristic(A)$ amounts to reduce the number of next atoms to select and makes the conflicts detection faster.

Algorithm 1. $smodels(A)$

```
A ← expand(A)
if conflict(A) then
  return false
else if A covers atom(E) then
  return true
else
  x ← heuristic(A)
  if smodels(A ∪ {x}) then
    return true
  else
    return smodels(A ∪ {not x})
  end if
end if
```

2.4 Removed Sets Revision

We briefly recall the Removed Sets Revision approach. The Removed Sets Revision [16] deals with the revision of a set of propositional formulae by a set of propositional formulae [1]. Let K and A be finite sets of clauses. The Removed Sets Revision focuses on the minimal subsets of clauses to remove from K, called *removed sets* [15], in order to restore the consistency of $K \cup A$. More formally:

Definition 2. *Let K and A be two consistent sets of clauses such that $K \cup A$ is inconsistent. R a subset of clauses of K, is a* removed set *of $K \cup A$ iff* (i) $(K \backslash R) \cup A$ *is consistent;* (ii) $\forall R' \subseteq K$, *if* $(K \backslash R') \cup A$ *is consistent then* $\mid R \mid \leq \mid R' \mid$[2].

Let denote by $\mathcal{R}(K \cup A)$ the collection of removed sets of $K \cup A$, the Removed Sets Revision (RSR) is defined as follows:

Definition 3. *Let K and A be two consistent sets of clauses. The removed sets revision is defined by:* $K \circ_{RSR} A =_{def} \bigvee_{R \in \mathcal{R}(K \cup A)} Cn((K \backslash R) \cup A)$.

[1] From now on, we consider propositional formulae in their equivalent conjunctive normal form (CNF).

[2] $\mid R \mid$ denotes the number of clauses of R.

3 Removed Sets Fusion

We propose a new syntactic fusion framework, Removed Set Fusion (RSF), which aims at merging several consistent belief bases. The approach consists in removing subsets of clauses from the union of the belief bases, according to a given strategy P in order to restore consistency. This framework captures the classical fusion operators and can be efficiently implemented. It generalizes the previously recalled RSR belief revision operation and requires a generalization of the notion of removed set.

Let $E = \{K_1, \ldots, K_n\}$ be a belief profile where K_i, $1 \leq i \leq n$ is a consistent belief base and let X and X' be two subsets of $K_1 \sqcup \ldots \sqcup K_n$.

Definition 4. *Let $E = \{K_1, \ldots, K_n\}$ be a belief profile such that $K_1 \sqcup \ldots \sqcup K_n$ is inconsistent, $X \subseteq K_1 \sqcup \ldots \sqcup K_n$ is a* potential removed set *of E iff $(K_1 \sqcup \ldots \sqcup K_n) \backslash X$ is consistent.*

The number of the potential removed sets is exponential with respect to the number of clauses in E. Hence, only the most relevant potential removed sets, according to a the chosen strategy, have to be selected. Therefore, a preference relation according to any strategy P, denoted by $\leq_P$, is defined and $X \leq_P X'$ means that X is preferred to X' according to P.

Definition 5. *Let $E = \{K_1, \ldots, K_n\}$ be a belief profile such that $K_1 \sqcup \ldots \sqcup K_n$ is inconsistent, $X \subseteq K_1 \sqcup \ldots \sqcup K_n$ is a* removed set *of E according to P iff*

1. *X is a potential removed set of E;*
2. *There is no $X' \subseteq K_1 \sqcup \ldots \sqcup K_n$ such that $X' <_P X$.*

We denote by $\mathcal{F}_P\mathcal{R}(E)$ the collection of removed sets [3] of E according to P, the Removed Set Fusion (RSF) is defined as follows.

Definition 6. *Let $E = \{K_1, \ldots, K_n\}$ be a belief profile. The fusion operation $\Delta^P(E)$ is defined by:*

$$\Delta^P(E) = \bigvee_{X \in \mathcal{F}_P\mathcal{R}(E)} \{Cn((K_1 \sqcup \ldots \sqcup K_n) \backslash X)\}$$

Classical merging operators are easy to use in this context, by instanciating the preceding definitions with the preference relations defined next.

3.1 Representing Classical Fusion Operations with RSF

We here define some of the classical merging operators ($Card$,Σ, Max, $Gmax$,...). They can be encoded by preference relations over potential removed sets.

[3] If $K_1 \sqcup \ldots \sqcup K_n$ is consistent $\mathcal{F}_P\mathcal{R}(E) = \emptyset$.

***Card* operation.** The *Card* operation is captured within our framework as follows:

Definition 7. *Let X and X' be two potential removed sets of E without repetition: $X \leq_{Card} X'$ iff $|X| \leq |X'|$.*

The *Card* strategy minimizes the number of clauses to remove from E and does not take repetitions into account. It is close to the *Comb*4 operator defined in [1].

***Σ* operation.** The Σ operation is captured within our framework as follows:

Definition 8. *Let X and X' be two potential removed sets of E: $X \leq_{\Sigma} X'$ iff $\sum_{1 \leq i \leq n} |X \cap K_i| \leq \sum_{1 \leq i \leq n} |X' \cap K_i|$.*

The Σ strategy minimizes the number of clauses to remove from E taking repetitions into account. It corresponds to the *intersection operator* defined in [25].

***Max* operation.** The *Max* operation is captured within our framework as follows:

Definition 9. *Let X and X' be two potential removed sets of E: $X \leq_{max} X'$ iff*
$\max_{1 \leq i \leq n} |X \cap K_i| \leq \max_{1 \leq i \leq n} |X' \cap K_i|$ *and* $X \subseteq X'$.

The *Max* strategy tries to spread the clauses to remove over the belief bases of E and minimizes the number of clauses to remove from the belief base the most involved in the inconsistency.

***Gmax* operation.** The *Gmax* operation is captured within our framework as follows:

Definition 10. *For each potential removed sets X and each belief base K_i, we define $p_X^i = |X \cap K_i|$. Let L_X^E be the sequence $(p_X^1, \ldots, p_X^n)$ given in a decreasing order. Let X and X' be two potential removed sets of E: $X \leq_{Gmax} X'$ iff $L_X^E <_{lex} L_{X'}^E$* [4].

The *GMax* strategy is a refinement of the *Max* strategy it removes clauses from the belief bases according to a decreasing order on the number of clauses involved in the inconsistency.

Example. We use the following example [2] to illustrate our framework. Consider the following situation : a teacher asks to his students which among the following languages SQL (denoted by s), O_2 (denoted by o) and Datalog (denoted by d) they would like to learn. The first one wants to learn SQL or O_2 but not Datalog ($K_1 = \{\neg d, s \vee o\}$). The second one wants to learn only *Datalog* or O_2but not both $K_2 = \{\neg s, d \vee o, \neg o \vee \neg d\}$. The third one wants to learn all three

[4] We denote by $<_{lex}$ the lexicographic order.

$K_3 = \{s, d, o\}$. Let $E = \{K_1 \sqcup K_2 \sqcup K_3\}$ be the corresponding belief profile. In this case, the result of the fusion will be :

- $\mathcal{F}_{Card}\mathcal{R}(E) = \{\{\neg s, d\}, \{s, d\}\}$ and $\Delta^{Card}(E) = \{\{\neg d, s \vee o, d \vee o, \neg o \vee \neg d, s, o\}, \{\neg d, s \vee o, \neg s, d \vee o, \neg o \vee \neg d, o\}\}$
- $\mathcal{F}_{\Sigma}\mathcal{R}(E) = \mathcal{F}_{Card}\mathcal{R}(E)$ and $\Delta^{\Sigma}(E) = \Delta^{Card}(E)$
- $\mathcal{F}_{Max}\mathcal{R}(E) = \{\{\neg s, d\}\}$ and $\Delta^{Max}(E) = \{\{\neg d, s \vee o, d \vee o, \neg o \vee \neg d, s, o\}\}$
- $\mathcal{F}_{GMax}\mathcal{R}(E) = \mathcal{F}_{Max}\mathcal{R}(E)$ and $\Delta^{GMax}(E) = \Delta^{Max}(E)$

We now present an implementation of RSF for the $Card$ and Σ strategies.

4 Encoding RSF in Answer Sets Programming

We now show how we construct a logic program, denoted by P_E, such that the preferred answer sets of P_E correspond to the removed sets of E.

We first show how to translate the Removed Set Fusion into a logic program, in the spirit of Niemelä in [20], in order to obtain a one-to-one correspondence between answer sets of P_E and potential removed sets of E. The key idea of the translation is to introduce for each clause, an atom which presence in the answer set corresponds to the presence of the clause in a potential removed set. We then define the notion of preferred answer set in order to perform RSF.

4.1 Translation into a Logic Program

Let $E = \{K_1, \ldots, K_n\}$ be an belief profile. The set of all positive literals of P_E is denoted by V^+. The set of all negative literals of P_E is denoted by V^-. The set of all atoms representing clauses is defined by $R^+ = \{r_c^i \mid c \in K_i\}$ and $CL(r_c^i)$ denotes the clause of K_i corresponding to r_c^i in P_E, namely $\forall r_c^i \in R^+,\ CL(r_c^i) = c$. To each answer set S of P_E, we associate the potential removed set $CL(R^+ \cap S)$.

1. The first step introduces rules in order to build a one-to-one correspondence between answer sets of P_E and interpretations of V^+. For each atom, $a \in V^+$ we introduce two rules : $a \leftarrow not\ a'$ and $a' \leftarrow not\ a$ where $a' \in V^-$ is the negative atom corresponding to a.
2. The second step excludes answer sets S which correspond to interpretations which are not models of $(K_1 \sqcup \ldots \sqcup K_n) \backslash C_i$ with $C_i = \{c | r_c \in S\}$. For each clause c of K_j such that $c = \neg b_o \vee \ldots \vee \neg b_n \vee b_{n+1} \vee \ldots \vee b_m$, we introduce the following rule $r_c^j \leftarrow b_o, \ldots, b_n, b'_{n+1}, \ldots, b'_m$

This translation differs from the one proposed in [26] for RSR since we only consider the positive atoms R^+ representing the clauses.

Example. The logic program P_E corresponding to the previous example is:

$$\begin{array}{ccc} s \leftarrow not\ s' & s' \leftarrow not\ s & d \leftarrow not\ d' \\ d' \leftarrow not\ d & o \leftarrow not\ o' & o' \leftarrow not\ o \\ r^1_{\neg d} \leftarrow d & r^1_{s \vee o} \leftarrow s', o' & r^2_{\neg s} \leftarrow s \\ r^2_{d \vee o} \leftarrow d', o' & r^2_{\neg d \vee \neg o} \leftarrow d, o & r^3_s \leftarrow s' \\ r^3_d \leftarrow d' & r^3_o \leftarrow o' & \end{array}$$

Let S be a set of atoms, we define I_S such that $I_S = \{a | a \in S\} \cup \{\neg a | a' \in S\}$. The following proposition establishes the correspondence between answer sets and models of $(K_1 \sqcup \ldots \sqcup K_n) \backslash CL(R^+ \cap S)$.

Proposition 1. *Let $E = \{K_1, \ldots, K_n\}$ be an belief profile. Let $S \subseteq V$ be a set of atoms. S is an answer set of P_E iff I_S is an interpretation of V^+ which satisfies $(K_1 \sqcup \ldots \sqcup K_n) \backslash CL(R^+ \cap S)$.*

In order to compute the answer sets corresponding to the removed sets we introduce the notion of preferred answer set according to a strategy P.

Definition 11. *Let P_E be a logic program and let S and S' be two set of atoms of P_E. S is a preferred answer set of P_E according to a strategy P iff*

- *S is an answer set of P_E;*
- *for every answer set S' of P_E, S' is not preferred to S according to P.*

The correspondence between preferred answer sets and removed sets is given by the following proposition for the strategies $Card$ and Σ.

Proposition 2. *Let $E = \{K_1, \ldots, K_n\}$ be an belief profile. X is a removed set of E according to the strategy P iff there exists a preferred answer set S of P_E according to P such that $CL(R^+ \cap S) = X$.*

Example. Let P_E be the logic program of the previous example. The collection of preferred answer sets of P_E according to the strategies $Card$ and Σ is : $\{S_1 = \{s, d', o, r^2_{\neg s}, r^3_d\}, S_2 = \{s', d', o, r^3_s, r^3_d\}\}$. Since $R^+ = \{r^1_{\neg d}, r^1_{s \vee o}, r^2_{\neg s}, r^2_{d \vee o}, r^2_{\neg d \vee \neg o}, r^3_s, r^3_d, r^3_o\}$, the removed sets are $CL(R^+ \cap S_1) = \{\neg s, d\}$ and $CL(R^+ \cap S_2) = \{s, d\}$.

4.2 Computing the Preferred Answer Sets : The RSF Algorithm

The RSF algorithm computes the preferred answer sets corresponding to the removed sets. This algorithm is a modification of Smodels algorithm that selects the preferred answer sets according to a chosen strategy P. It builds, step by step, a collection of candidate answer sets. At the end of the computation, this collection contains all the preferred answer sets corresponding to the removed sets.

The selection of the preferred answer sets is achieved thanks to the function $Condition_P(A)$, where A is a set of atoms. This function compares the current answer set candidate to the preferred answer sets previously computed. The three possible behaviors of the function $Condition_P(A)$ are:

1. A cannot lead to any preferred answer set. In this case, the computation is stopped and the algorithm backtracks;
2. A is complete and is equally preferred to the previously computed best answer sets. In this case, A is added to the collection of candidates answer sets;

3. A is complete and is preferred to the previously computed best answer sets. In this case, the collection reduced to A replaces the collection of candidates answer sets.

Another adaptation of Smodels concerns the initial heuristic of Smodels. If an atom a is selected then the atom a' cannot be deduced anymore. The only atoms that can be deduced are atoms that represent the rules r_c^i. The use of the standard heuristic leads to maximize the number of deduced r_c^i which contradicts the objective of RSF. It doesn't allow us to take advantage of the pruning of the search tree. We modify the initial heuristic in order to select the atoms that minimize the number of deduced atoms. Therefore, the first computed answer sets have greater chances to be preferred according to the chosen strategy. The new function is called $mheuristic(A)$.

Algorithm 2. Algorithm $rsf(A)$

```
A ← expand(A)
if conflict(A) then
   return false
end if
if (1) Condition_P(A) = 1 then
   return false
else if A is a subset of an already computed model then
   return false
else if A covers atom(E) then
   if (2) Condition_P(A) = 0 then
      add A to the set of solutions
      return true
   else
      (3){A} becomes the set of solutions
      return true
   end if
end if
x ← mheuristic(A)
rsf(A ∪ {x})
rsf(A ∪ {not x})
```

The adaptations of the original Smodels algorithm consist in: (i) avoiding all the subsets of R^+ leading to answer sets which removes more clauses than the removed sets; (ii) not computing several times the same subsets of literals of R^+; (iii) taking advantage of possible cuts in the search space.

5 Preliminary Experimental Study

We now present the results of a preliminary experimental study of the RSF approach. The tests were conducted on a Centrino cadenced at 1.73GHz and equipped with 1GB of RAM.

As far as we know, there is no other implementation of the fusion of propositional belief bases nor benchmarks for fusion. The following preliminary tests are not exhaustive enough to conclude about the efficiency of RSF. Nevertheless, they show the practicability of the approach. In order to be able to conclude on the efficiency of RSF, we plan to develop a more complete set of tests.

Table 1. Results for $nb = 3$, $sc = 3$ and $d = 20\%$

nc	nv	$Succes(\%)$	$Time(s)$
100	1000	100	2, 1
200	2000	100	7, 2
400	4000	100	37, 6

nc	nv	$Succes(\%)$	$Time(s)$
600	6000	100	105, 2
800	8000	100	221, 4
1200	12000	0	–

Table 2. Results for $nb = 3$, $sc = 3$ and $d = 20\%$ according to nv/nc

nc	nv	$Succes(\%)$	$Time(s)$
400	200	40	68, 7
400	400	20	13, 5
400	800	70	17, 5

nc	nv	$Succes(\%)$	$Time(s)$
200	100	90	2, 2
200	200	90	11, 1
200	400	90	2, 1

Benchmarks are randomly generated according to several parameters: the number of bases (nb), the number of clauses in the bases (nc), the number of variables in the bases (nv), the size of clauses in the bases (sc) and a parameter that measures how belief bases differ (d).

Test bases are constructed as follows. We randomly construct an interpretation and then we randomly generate clauses that satisfy it. From one base to another, we change this interpretation according to the parameter(d) which represents the percentage of changed variables. For each set of parameters, we launched 10 different sets of the test bases. A test is considered successfull if it computes all the removed sets in less than 300 seconds and we keep the average run time of the successfull tests. The experimentation gives the percentage of successfull tests and the run time, in seconds, for the computation of all removed sets.

Table 1 shows the result of the behaviour of RSF algorithm for 3 bases consisting on ternary clauses. The RSF approach performs the fusion of 3 bases with a reasonable run time until a total number of about 3000 clauses and 8000 variables. Making nv/nc vary, table 2 shows a peak of difficulty when nv/nc approaches 1.

Analysing the running time we have observed that the heuristic for choosing the atoms is time consuming and has to be improved.

6 Conclusion

This paper presents a new approach for performing syntactic fusion of propositional beliefs bases and shows that the classical fusion operations $Card$, Σ, Max, $GMax$, initially defined at the semantic level, can be expressed within this syntactic framework.

The paper shows that RSF can be successfully encoded into answer set programming for the strategies $Card$ and Σ and proposes an implementation stemming from Smodels system. It presents a preliminary experimental study that seems promising for performing belief bases fusion on real scale applications. A future work will conduct the implementation of Max and $GMax$ strategies.

A more extensive experimentation has to be conducted on real scale applications in order to provide a more accurate evaluation of the performance of RSF. This will be conducted in a future work in the framework of an european project in the context of fusion of spatial information. Moreover, the development of a benchmarking platform for fusion will be useful, not only for testing RSF, but more globally for anyone willing to work on practical implementations of fusion operations.

Removed Set Fusion (RSF) makes it possible to efficiently implement the classical $Card$ and Σ fusion operators, moreover it generalizes Removed Set Revision (RSR) since belief bases revision can be considered as the prioritized merging of two belief bases [27] and RSR amounts to the fusion of two sources according to the $Card$ strategy.

Our framework could be extended according to several directions. A first extension for dealing with constraints that the merged belief base $\Delta(E)$ has to satisfy. A second extension to prioritized belief bases fusion.

A future work will detail the semantic characterization of RSF. This characterization is provided from the set of clauses of $K_1 \sqcup \cdots \sqcup K_n$ falsified by an interpretation.

Acknowledgements

Work partially supported by the European Community under project VENUS (Contract IST-034924) of the "Information Society Technologies (IST) program of the 6th FP for RTD". The authors are solely responsible for the content of this paper. It does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of data therein.

References

1. Baral, C., Kraus, S., Minker, J., Subrahmanian, V.S.: Combining knowledge bases consisting of first order theories. In: ISMIS, pp. 92–101 (1991)
2. Revesz, P.Z.: On the semantics of theory change: arbitration between old and new information. In: 12th ACM symp. on Principles of Databases, pp. 71–92 (1993)
3. Lin, J.: Integration of weighted knowledge bases. AI 83, 363–378 (1996)
4. Revesz, P.Z.: On the semantics of arbitration. J. of Alg. and Comp. 7(2), 133–160 (1997)
5. Cholvy, L.: Reasoning about merging information. Handbook of Defeasible Reasoning and Uncertainly Management Systems 3, 233–263 (1998)
6. Konieczny, S., Pérez, R.P.: On the logic of merging. In: Proc. of KR'98, pp. 488–498 (1998)
7. Lafage, C., Lang, J.: Logical representation of preferences for group decision making. In: Proc. of KR'00, Breckenridge, pp. 457–468 (2000)
8. Konieczny, S.: On the difference between merging knowledge bases and combining them. In: Proc. of KR2000, Breckenridge, pp. 135–144. Morgan Kaufmann, San Francisco (2000)

9. Delgrande, J., Dubois, D., Lang, J.: Iterated revision as prioritized merging. In: Proc. of KR06, Lake District (UK), pp. 210–220. AAAI Press, Stanford (2006)
10. Fagin, R., Kuper, G.M., Ullman, J.D., Vardi, M.Y.: Updating logical databases. Advances in Comp. Research, 1–18 (1986)
11. Meyer, T., Ghose, A., Chopra, S.: Syntactic representations of semantic merging operations. In: IJCAI'01 Workshop on Incons. in Data and Knowledge (2001)
12. Konieczny, S., Lang, J., Marquis, P.: Distance-based merging: A general framework and some complexity results. In: Proc. of KR'02, pp. 97–108 (2002)
13. Dubois, D., Lang, J., Prade, H.: Possibilistic Logic. Handbook of Logic in Artificial Intelligence and Logic Programming 3, 439–513 (1994)
14. Benferhat, S., Dubois, D., Kaci, S., Prade, H.: Possibilistic Merging and Distance-based Fusion of Propositional Information. AMAI'02 34(1-3), 217–252 (2002)
15. Papini, O.: A complete revision function in propositionnal calculus. In: Proc. of ECAI92, pp. 339–343. J. Wiley and Sons, Chichester (1992)
16. Würbel, E., Jeansoulin, R., Papini, O.: Revision: An application in the framework of gis. In: Proc. of KR'00, Breckenridge, pp. 505–516 (2000)
17. Eiter, T., Leone, N., Mateis, C., Pfeifer, G., Scarcello, F.: The kr system dlv: progress report, comparison and benchmarks. In: Proc. of KR'98, pp. 406–417 (1998)
18. Cholewinski, P., Marek, V., Mikitiuk, A., Truszczynski, M.: Computing with default logic. AI 112, 105–146 (1999)
19. Rao, P., Sagonas, K., Swift, W.D.S., Friere, J.: Xsb: A system for efficiently computing well-founded semantics. In: Fuhrbach, U., Dix, J., Nerode, A. (eds.) LPNMR 1997. Lecture Notes in Computer Science(LNAI), vol. 1265, pp. 430–440. Springer, Heidelberg (1997)
20. Niemelä, I., Simons, P.: An implementation of stable model and well-founded semantics for normal logic programs. In: Fuhrbach, U., Dix, J., Nerode, A. (eds.) LPNMR 1997. LNCS, vol. 1265, pp. 420–429. Springer, Heidelberg (1997)
21. Linke, T.: More on nomore. In: Proc. of NMR'02 (2002)
22. Lin, J., Mendelzon, A.O.: Merging databases under constraints. IJCIS'98 7(1), 55–76 (1998)
23. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proc. of the Fifth Int. Conf. on Logic Prog., pp. 1070–1080. MIT Press, Cambridge (1988)
24. Simons, P.: Extending and implementing the stable model semantics (2000)
25. Konieczny, S.: La logique du changement - Revision et fusion de connaissances. PhD thesis, Univ. de Lille (1999)
26. Bennaim, J., Benferhat, S., Papini, O., Würbel, E.: An answer set programming encoding of prioritized removed sets revision: application to gis. In: Alferes, J.J., Leite, J.A. (eds.) JELIA 2004. LNCS (LNAI), vol. 3229, pp. 604–616. Springer, Heidelberg (2004)
27. Delgrande, J., Dubois, D., Lang, J.: Iterated revision as prioritized merging. In: Proc. of KR'06, Windermere, pp. 210–220 (2006)

COBA 2.0: A Consistency-Based Belief Change System

James P. Delgrande[1], Daphne H. Liu[1], Torsten Schaub[2,⋆], and Sven Thiele[2]

[1] School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6
[2] Institut für Informatik, Universität Potsdam, August-Bebel-Str. 89, D-14482 Potsdam, Germany

Abstract. We describe COBA 2.0, an implementation of a consistency-based framework for expressing belief change, focusing here on revision and contraction, with the possible incorporation of integrity constraints. This general framework was first proposed in [1]; following a review of this work, we present COBA 2.0's high-level algorithm, work through several examples, and describe our experiments. A distinguishing feature of COBA 2.0 is that it builds on SAT-technology by using a module comprising a state-of-the-art SAT-solver for consistency checking. As well, it allows for the simultaneous specification of revision, multiple contractions, along with integrity constraints, with respect to a given knowledge base.

1 Introduction

Given a knowledge base and a sentence for revision or contraction, the fundamental problem of belief change is to determine what the resulting knowledge base contains. The ability to change one's knowledge is essential for an intelligent agent. Such change in response to new information is not arbitrary, but rather is typically guided by various rationality principles. The best known of these sets of principles was proposed by Alchourron, Gardenfors, and Makinson [2], and has come to be known as the AGM approach.

In this paper, we describe COBA 2.0, an implementation of a consistency-based approach to belief revision and contraction. The general methodology was first proposed in [1]. In this approach, the AGM postulates for revision are effectively satisfied, with the exception of one of the "extended" postulates. Similarly the contraction postulates are satisfied with the exception of the controversial recovery postulate and one of the extended postulates. Notably the approach is syntax independent, and so independent of how a knowledge base and sentence for belief change is represented. COBA 2.0 implements this approach, and in a more general form. Thus a single belief change operation will involve a single knowledge base and (possibly) a sentence for revision, but along with (possibly) a set of sentences for contraction; as well integrity constraints are handled, and in a straightforward fashion.

In Section 2, we give background terminology, notation, and implementation considerations. Section 3 presents COBA 2.0's high-level algorithm, in addition to working through two examples. Section 4 discusses COBA 2.0's features, syntax, and input checks, while Section 5 describes our experiments evaluating COBA 2.0 against a comparable solver. Lastly, Section 6 concludes with a summary.

⋆ Affiliated with Computing Science at Simon Fraser University and IIIS at Griffith University.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 78–90, 2007.

2 Preliminaries

To set the stage, we informally motivate our original approach to belief revision; contraction is motivated similarly, and is omitted here given space considerations. First, the syntactic form of a sentence doesn't give a clear indication as to which sentences should or should not be retained in a revision. Alternately, one can consider interpretations, and look at the models of K and α. The interesting case occurs when $K \cup \{\alpha\}$ is unsatisfiable because K and α share no models. Intuitively, a model of $K\dot{+}\alpha$ should then contain models of α, but incorporating "parts" of models of K that don't conflict with those of α. That is, we will have $Mod(K\dot{+}\alpha) \subseteq Mod(\alpha)$, and for $m \in Mod(K\dot{+}\alpha)$ we will want to incorporate whatever we can of models of K.

We accomplish this by expressing K and α in different languages, but such that there is an isomorphism between atomic sentences of the languages. In essence, we replace every occurrence of an atomic sentence p in K by a new atomic sentence p', yielding knowledge base K' and leaving α unchanged. Clearly, under this relabelling, the models of K' and α will be independent, and $K' \cup \{\alpha\}$ will be satisfiable (assuming that each of K, α are satisfiable). We now assert that the languages agree on the truth values of corresponding atoms wherever consistently possible. So, for every atomic sentence p, we assert that $p \equiv p'$ whenever this is consistent with $K' \cup \{\alpha\}$ along with the set of equivalences obtained so far. We obtain a maximal set of such equivalences, call it EQ, such that $K' \cup \{\alpha\} \cup EQ$ is consistent. A model of $K' \cup \{\alpha\} \cup EQ$ then will be a model of α in the original language, wherein the truth values of atomic sentences in K' and α are linked via the set EQ. A candidate "choice" revision of K by α consists of $K' \cup \{\alpha\} \cup EQ$ re-expressed in the original language. General revision corresponds to the intersection of all candidate choice revisions. The following section gives an example, once we have given a formal summary of the approach.

2.1 Formal Preliminaries

We deal with propositional languages and use the logical symbols $\top$, $\bot$, $\neg$, $\vee$, $\wedge$, $\supset$, and $\equiv$ to construct formulas in the standard way. We write $\mathcal{L}_{\mathcal{P}}$ to denote a language over an alphabet $\mathcal{P}$ of propositional letters or atomic propositions. Formulas are denoted by the Greek letters α, β, α_1, ... Knowledge bases, identified with belief sets or deductively-closed sets of formulas, are denoted by K, K_1, ... So $K = Cn(K)$, where $Cn(\cdot)$ is the deductive closure in classical propositional logic of the formula or set of formulas given as argument. Given an alphabet $\mathcal{P}$, we define a disjoint alphabet $\mathcal{P}'$ as $\mathcal{P}' = \{p' \mid p \in \mathcal{P}\}$. For $\alpha \in \mathcal{L}_{\mathcal{P}}$, α' is the result of replacing in α each proposition $p \in \mathcal{P}$ by the corresponding proposition $p' \in \mathcal{P}'$ (and hence an isomorphism between $\mathcal{P}$ and $\mathcal{P}'$). This definition applies analogously to sets of formulas.

A *belief change scenario* in $\mathcal{L}_{\mathcal{P}}$ is a triple $B = (K, R, C)$ where K, R, and C are sets of formulas in $\mathcal{L}_{\mathcal{P}}$. Informally, K is a belief set that is to be modified so that the formulas in R are contained in the result, and the formulas in C are not. An extension determined by a belief change scenario is defined as follows.

Definition 1 (Belief Change Extension). *Let $B = (K, R, C)$ be a belief change scenario in $\mathcal{L}_\mathcal{P}$, and a maximal set of equivalences $EQ \subseteq \{p \equiv p' \mid p \in \mathcal{P}\}$ be such that $Cn(K' \cup R \cup EQ) \cap (C \cup \{\bot\}) = \emptyset$.*

Then $Cn(K' \cup R \cup EQ) \cap \mathcal{L}_\mathcal{P}$ is a belief change extension *of B. If there is no such set EQ, then B is* inconsistent *and $\mathcal{L}_\mathcal{P}$ is defined to be the sole* (inconsistent) belief change extension *of B.*

In Definition 1, "maximal" is with respect to set containment, and the exclusive use of "$\{\bot\}$" is to take care of consistency if $C = \emptyset$. Definition 1 provides a very general framework for specifying belief change. Next, we can restrict the definition to obtain specific functions for belief revision and contraction.

Revision and Contraction. For a given belief change scenario, there may be more than one consistent belief change extension. We can thus use a *selection function* c that, for any set $I \neq \emptyset$, has as value some element of I.

Definition 2 (Revision). *Let K be a knowledge base, α a formula, and $(E_i)_{i \in I}$ the family of all belief change extensions of $(K, \{\alpha\}, \emptyset)$. Then, we define*

1. *$K \dot{+}_c \alpha = E_i$ as a* choice revision *of K by α with respect to some selection function c with $c(I) = i$.*
2. *$K \dot{+} \alpha = \bigcap_{i \in I} E_i$ as the* (skeptical) revision *of K by α.*

Definition 3 (Contraction). *Let K be a knowledge base, α a formula, and $(E_i)_{i \in I}$ the family of all belief change extensions of $(K, \emptyset, \{\alpha\})$. Then, we define*

1. *$K \dot{-}_c \alpha = E_i$ as a* choice contraction *of K by α with respect to some selection function c with $c(I) = i$.*
2. *$K \dot{-} \alpha = \bigcap_{i \in I} E_i$ as the* (skeptical) contraction *of K by α.*

A *choice* change represents a feasible way in which a knowledge base can be revised or contracted to incorporate new information. On the other hand, the intersection of all choice changes represents a "safe," *skeptical* means of taking into account all choice changes.

Table 1 gives examples of skeptical revision. The knowledge base is in the first column, but with atoms already renamed. The second column gives the revision formula, while the next lists the maximal consistent EQ set(s); the last column gives the results of the revision, as a finite representation of $Cn(K \dot{+} \alpha)$. For $\{p \wedge q\} \dot{+} (\neg p \vee \neg q)$, there are two maximal consistent EQ sets $\{p \equiv p'\}$ and $\{q \equiv q'\}$ and thus two corresponding

Table 1. Skeptical Revision Examples

K'	α	EQ	$K \dot{+} \alpha$
$p' \wedge q'$	$\neg q$	$\{p \equiv p'\}$	$p \wedge \neg q$
$\neg p' \equiv q'$	$\neg q$	$\{p \equiv p', q \equiv q'\}$	$p \wedge \neg q$
$p' \vee q'$	$\neg p \vee \neg q$	$\{p \equiv p', q \equiv q'\}$	$p \equiv \neg q$
$p' \wedge q'$	$\neg p \vee \neg q$	$\{p \equiv p'\}, \{q \equiv q'\}$	$p \equiv \neg q$

Table 2. Skeptical Contraction Examples

K'	α	EQ	$K \dot{-} \alpha$
$p' \wedge q'$	q	$\{p \equiv p'\}$	p
$p' \wedge q' \wedge r'$	$p \vee q$	$\{r \equiv r'\}$	r
$p' \vee q'$	$p \wedge q$	$\{p \equiv p', q \equiv q'\}$	$p \vee q$
$p' \wedge q'$	$p \wedge q$	$\{p \equiv p'\}, \{q \equiv q'\}$	$p \vee q$

choice extensions $Cn(p \wedge \neg q)$ and $Cn(\neg p \wedge q)$, respectively. Table 2 lists four skeptical contraction examples.

The general approach, with $|C| > 1$, can be employed to express *multiple contraction* [3], in which contraction is with respect to a set of (not necessarily mutually consistent) sentences. Therefore, we can use the belief change scenario $(K, \emptyset, \{\alpha, \neg\alpha\})$ to represent a *symmetric contraction* [4] of α from K. Refer to [1] for a discussion of the formal properties of these belief revision and contraction operators.

Integrity Constraints. Definition 1 allows for simultaneous revision and contraction by sets of formulas. This in turn leads to a natural and general treatment of integrity constraints. To specify a belief change incorporating a set of *consistency-based* integrity constraints [5,6], IC_c, and a set of formulas as entailment-based constraints [7], IC_e, one can specify a belief change scenario by $(K, R \cup IC_e, C \cup \overline{IC_c})$, where K, R, and C are as in Definition 1, and $\overline{IC_c} = \{\neg\phi \mid \phi \in IC_c\}$. See [1] for details.

2.2 Implementation Considerations

Finite Representation. Definitions 1–3 provide an abstract characterization of revision and contraction, yielding in each case a deductively-closed belief set. It is proven in [1] that the same (with respect to logical equivalence) operators can be defined so that they yield a knowledge base consisting of a finite formula. Consider $K \dot{+} \alpha$. Via Definitions 1 and 2, we determine maximal sets EQ where $\{K'\} \cup \{\alpha\} \cup EQ$ is consistent. For each such EQ set, we carry out the substitutions:

- for $p \equiv p' \in EQ$, replace p' with p in K',
- for $p \equiv p' \notin EQ$, replace p' with $\neg p$ in K'.

It is shown that following this substitution, the resulting knowledge base and input formula is logically equivalent to some choice revision; the disjunction of all such resulting knowledge bases and input formula is equivalent to the skeptical revision.

For contraction (where $C \neq \emptyset$), we need to substitute into the resulting K all possible combinations of truth value assignments for all elements in $P_{\overline{EQ}}$. Again, see [1] for details.

Limiting Range of EQ. The range of EQ can be limited to "relevant" atoms. Intuitively, if an atomic sentence appears in a knowledge base K but not in the sentence for revision α, or vice versa, then that atomic sentence plays no part in the revision process. The same intuition extends to contraction. It was proven in [1] that for computing a belief change extension of a belief change extension $B = (K, R, C)$, we need consider only

those atoms common to K and to $(R \cup C)$. That is, if $Atoms(X)$ is the set of atoms in set of formulas X, then in Definition 1 for forming K' and the set EQ we can limit ourselves to considering atoms in $Atoms(K) \cap (Atoms(R) \cup Atoms(C))$.

3 Algorithm

The results at the end of the last section lead to an algorithm for computing a belief change extension for an arbitrary belief change scenario. After presenting our algorithm, we will work through two example belief change scenarios.

Given a set K of formulas in $\mathcal{L}_\mathcal{P}$, and sets Rev, IC_e, Con, and IC_c of formulas in $\mathcal{L}_\mathcal{P}$ for revision, entailment-based integrity constraints, contraction, and consistency-based integrity constraints, respectively, algorithm $ComputeBCE$ returns a formula whose deductive closure is a belief change extension of the belief change scenario $B = (K, Rev \cup IC_e, Con \cup \overline{IC_c})$, where $\overline{IC_c} = \{\neg\phi \mid \phi \in IC_c\}$.

Algorithm $ComputeBCE$ invokes the following auxiliary functions:

$Atoms(S)$ Returns the set of atoms appearing in any formula in set of formulas S.

$Prime(K, CA)$ For set of formulas K and set of atoms CA, returns K but where every atom $p \in A$ is replaced by p'.

$Initialize(K', R, Con, IC_c)$ Given a formula K' and sets R, Con, IC_c of formulas, returns a set of formulas of form $(K' \wedge (\bigwedge R) \wedge \neg\phi \wedge \psi)$, for each $\phi \in (Con \cup \{\bot\})$ and $\psi \in (IC_c \cup \{\top\})$.

$Replace(K', p', p)$ Returns K' with every occurrence of atom p' replaced by p.

$ForgetOutEquiv(K', Out)$ Input: formula K' and a set Out of equivalences of atoms

Output: K' with every atom p such that $(p' \equiv p) \in Out$ is "forgotten":

1. If $Out = \emptyset$, then return K'.
2. $OutAtoms := \{p \mid (p' \equiv p) \in Out\}$.
3. $TA := PowerSet(OutAtoms)$.
   //TA is the set of all truth assignments to $OutAtoms$.
4. $KDisj := \bot$.
5. For each truth assignment $\pi \in TA$ {
   $TempK := K'$.
   $KDisj := KDisj \vee Substitute(TempK, \pi)$. }
   //$Substitute$ returns π applied to $TempK$.
6. Return $KDisj$.

Algorithm $ComputeBCE(K, Rev, IC_e, Con, IC_c)$
Let $R = Rev \cup IC_e$ and $C = Con \cup \overline{IC_c}$.
1. If $R \vdash \bot$ or $K \vdash \bot$, then return $\bot$.
2. If (for any $\psi \in IC_c$, $R \cup \{\psi\} \vdash \bot$), then return $\bot$.
3. If (for any $\phi \in Con$, $R \cup \{\neg\phi\} \vdash \bot$), then return $\bot$.
4. If (for any $\phi \in Con$ and any $\psi \in IC_c$
   $\{\neg\phi\} \cup \{\psi\} \vdash \bot$), then return $\bot$.
5. $CA := Atoms(K) \cap (Atoms(R) \cup Atoms(C))$.

6. $K' := Prime(K, CA)$.
7. $KR\overline{C} := Initialize(K', R, Con, IC_c)$.
8. $In := Out := \emptyset$.
9. For each $e \in \{p' \equiv p \mid p \in CA\}$ {
   If (for any $\theta \in KR\overline{C}$ we have $e \cup \{\theta\} \vdash \bot$)
   Then $Out := Out \cup \{e\}$.
   Else $In := In \cup \{e\}$. }
10. For each $e \in In$: $K' := Replace(K', p', p)$.
11. For each $e \in Out$: $K' := Replace(K', p', \neg p)$.
12. If ($Con \neq \emptyset$) Then $K' := ForgetOutEquiv(K', Out)$.
13. Return $K' \wedge (\bigwedge Rev)$.

Algorithm $ComputeBCE$ generates a belief change extension in non-deterministic polynomial (NP) time; i.e., an extension can be computed by a deterministic polynomial Turing machine using the answers given by an NP oracle. For this purpose, we currently use the SAT-solver called Berkmin in the SAT4J library [8]. The solver performs the consistency checks in lines 1 through 4 and within the for loop in Line 9. Before passing any formula to the solver, we convert it first to conjunctive normal form (CNF). The CNF formula, once created, is saved with its corresponding formula so that conversions are not done repetitively.

The selection function (for the "preferred" EQ set) is left implicit in Line 9 of Algorithm $ComputeBCE$; it is realized by the particular order chosen when treating the atoms in CA. In COBA 2.0, however, we create an ordered (in ascending cardinality) list L of all $2^{|CA|}$ possible subsets of $\{p' \equiv p \mid p \in CA\}$. To help streamline the search for EQ sets and minimize memory usage, we represent each equivalence by a single bit so that it is included in an EQ set e iff its corresponding bit is 1 in e's bit-string. Furthermore, the ordered list L can accommodate our subsequent search for maximal EQ sets, whether the search be breadth-first or depth-first. On average, the running time and memory usage of breadth-first search is comparable to that of depth-first search, although in our experience neither is consistently superior.

3.1 Examples

We illustrate how COBA 2.0 computes belief change extensions by working through two examples. The examples include belief revision and contraction.

Revision. Consider revising a knowledge base $K = \{p, q\}$ by a formula $\alpha = \neg p \vee \neg q$. We show how COBA 2.0 computes $K \dot{+} \alpha$:

1. Find the common atoms between the knowledge base and the revision formula.
   $CA = \{p, q\}$
2. Create a new formula K' from K by priming the common atoms appearing in K.
   $K' = (p' \wedge q')$
3. Find all maximal equivalence sets $EQ = \{b' \equiv b \mid b \in CA\}$ such that $\{K'\} \cup \{\alpha\} \cup EQ$ is satisfiable.
   $EQ_1 = \{p' \equiv p\}$
   $EQ_2 = \{q' \equiv q\}$

4. For each EQ_i, create a belief change extension by (a) unpriming in K' every primed atom p' if $(p' \equiv p) \in EQ_i$, (b) replacing every primed atom p' with $\neg p$ if $(p' \equiv p) \notin EQ_i$, and finally (c) conjoining K' with the revision formula.
   $K \dot{+}_{c_1} \{\alpha\} = (p \wedge \neg q) \wedge (\neg p \vee \neg q) \equiv (p \wedge \neg q)$
   $K \dot{+}_{c_2} \{\alpha\} = (\neg p \wedge q) \wedge (\neg p \vee \neg q) \equiv (\neg p \wedge q)$
5. The resulting knowledge base is the deductive closure of either the disjunction of all belief change extensions for *skeptical* change, or one belief change extension for *choice* change.
   $K \dot{+} \{\alpha\} = Cn((p \wedge \neg q) \vee (\neg p \wedge q))$

Contraction. Consider contracting a knowledge base $K = \{p \vee q\}$ by a formula $\alpha = p \vee q$. We show how COBA 2.0 computes $K \dot{-} \alpha$:

1. Find the common atoms between the knowledge base and the contraction formula.
   $CA = \{p, q\}$
2. Create a new formula K' from K by priming the common atoms appearing in K.
   $K' = (p' \vee q')$
3. Find all maximal equivalence sets $EQ = \{b' \equiv b \mid b \in CA\}$ such that $\{K'\} \cup \{\neg\alpha\} \cup EQ$ is satisfiable.
   $EQ_1 = \{\}$
4. For each EQ_i, create a belief change extension by (a) unpriming in K' every primed atom p' if $(p' \equiv p) \in EQ_i$, (b) replacing every primed atom p' with $\neg p$ if $(p' \equiv p) \notin EQ_i$, and finally (c) taking the disjunction of all possible substitutions of $\top$ or $\bot$ into those atoms in K' that are in CA but whose corresponding equivalences are not in EQ_i.
   $K \dot{-}_{c_1} \{\alpha\} = (\top)$
5. The resulting knowledge base is the deductive closure of either the disjunction of all belief change extensions for $skeptical$ change, or one belief change extension for $choice$ change.
   Here, there is only one resulting knowledge base for skeptical change and for choice change: $K \dot{-} \{\alpha\} = Cn((\neg\bot \vee \neg\bot) \vee (\neg\bot \vee \neg\top) \vee (\neg\top \vee \neg\bot) \vee (\neg\top \vee \neg\top)) = Cn(\top)$

4 Implementation

In this section, we describe the COBA 2.0 implementation. We discuss features, syntax, and syntactic and consistency checks on input formulas.

4.1 Features

COBA 2.0 is available as an interactive Java applet, complete with a menu, text boxes, buttons, and separate panels for belief change, integrity constraints, and snapshots. Via the menu, users can import belief change scenarios from files, specify the type (skeptical or choice) of belief change desired, and obtain a resulting knowledge base.

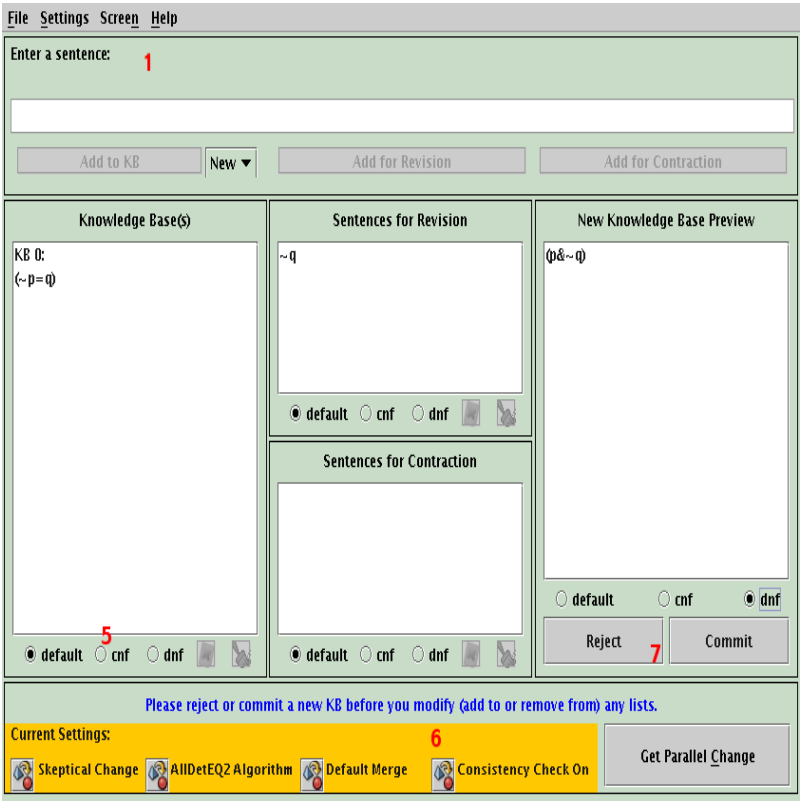


Fig. 1. COBA 2.0's Main Screen

Users may also

1. enter belief change scenarios in text boxes,
2. view logs of the changes made to the knowledge base (KB) list, the entailment-based integrity constraints (EB IC) list, and the consistency-based integrity constraints (CB IC) list,
3. revert to an older KB, EB IC, or CB IC snapshot,
4. save any list to an output file,
5. view formulas in CNF or DNF,
6. turn off the various consistency checks,
7. preview, and then reject or commit, a resulting knowledge base, and
8. view the user manual and JavaDocs in external browser windows (if the applet is running in an html document).

COBA 2.0 automatically simplifies formulas where applicable, for example, eliminating occurrences of $\top$ and $\bot$ in subformulas. COBA 2.0 also automatically informs users of any syntactically ill-formed input formulas. The consistency checks in 6. aboveand the syntax checks are elaborated on in Subsection 4.3. The applet, user manual, Java code, and Javadocs of COBA 2.0 are accessible from [9].

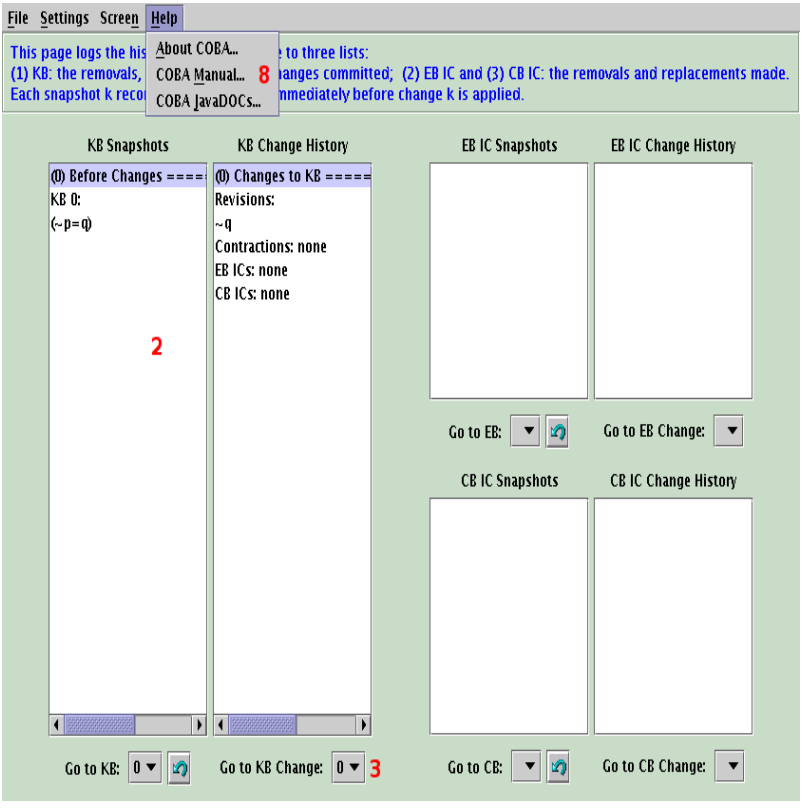


Fig. 2. COBA 2.0's History Screen

4.2 Syntax

COBA 2.0 accepts almost all alphanumerical strings for atom names. The exceptions are the symbols in the following list: `'`, `+`, `&`, `^`, `~`, `=`, `>`, (and). Note that `T` and `F` stand for $\top$ and $\bot$, respectively.

More complex formulas can be built from formulas A and B using connectives.

- `~A` for the negation of A
- `(A&B)` for the conjunction of A and B
- `(A+B)` for the disjunction of A and B
- `(A>B)` for A implies B
- `(A=B)` for A is equivalent to B

A top-level formula with a binary connective (`&`, `+`, `>`, or `=`) must be enclosed in parentheses. Parentheses within a formula, however, are optional and are used only to enforce precedence. For example, `(a&b+c)` is a valid input sentence and is different from `(a&(b+c))`, whereas a top-level sentence like `a&b` is syntactically ill formed.

Encoding Input Files. Input file formats (for belief change scenarios) vary according to the list (KB, Revision, Contraction, EB IC, or CB IC) to which formulas are to be added. Any KB file to be loaded should precede each knowledge base by a line "KB :" (without the double quotes) and list each formula on a separate line. Each formula is listed on a separate line in any Revision and EB IC input files. For any contraction and CB IC input files, each line is interpreted as an independent formula for contraction and as a CB IC, respectively.

Consider an example contraction file. While the formula `(p&~q)` means that $(p \& \neg q)$ is to be removed from the consequences of the resulting knowledge base, `p` / `~q` listed on two separate lines means that both p and $\neg q$ are to be dropped from the consequences of the resulting knowledge base.

As an example, the next table shows some valid input files.

KB	Rev	Cont	EB IC	CB IC
KB : (p&q&r) (~q+~s)	q ~p	p ~q	(a&b+c) (x&(y+z))	d ~d

4.3 Input Checks

COBA 2.0 performs syntax and consistency checks on all input formulas. The former checks are always enforced, while the latter checks are optional but carried out by default. See below for details.

Syntax Checks. With regard to the syntax detailed earlier in Subsection 4.2, COBA 2.0 informs users of ill-formed input formulas. Thus, for example, the following ill-formed input strings would be flagged with an appropriate message: `q)`, `q+`, `p^`, `p'`, `(p`, `(p&(q)`, `(p+q&)`, and `(+q)`.

Consistency Checks. To preempt inconsistent belief change scenarios, COBA 2.0 prohibits certain kinds of input formulas that result in inconsistent belief change scenarios. This preemptive measure accords well with the consistency checks in lines 1 through 4 of Algorithm $ComputeBCE$ in Section 3. Automatic consistency checks on input formulas, although carried out by default, can be optionally disabled by users wishing to speed up computations. One caveat is that, if these checks are disabled, `F` might be obtained as the resulting knowledge base.

Let $(\bigwedge Rev)$ denote the conjunction of all formulas in Rev for revision, $(\bigwedge EBIC)$ the conjunction of all entailment-based integrity constraints. The following inconsistent belief change scenarios should be avoided; sample error messages, where applicable, are italicised.

1. a contradiction in Rev: *The conjunction of revisions is inconsistent!*
2. a contradiction in $EBIC$: *The conjunction of EB ICs is a contradiction!*
3. a contradiction as a KB, revision, or EB IC formula: No error message; sentence not added.

4. a tautology as a contraction formula: No error message; sentence not added.
5. a contradiction as a CB IC formula: No error message; sentence not added.
6. conflict between $(\bigwedge Rev)$ and $(\bigwedge EBIC)$: *The conjunction of revisions is inconsistent with the conjunction of EB ICs!*
7. conflict between $(\bigwedge Rev)$ and contraction formulas: *The contraction indexed 0 is inconsistent with the conjunction of revisions (indexing starts at 0)!*
8. conflict between $(\bigwedge Rev)$ and CB IC formulas: *The CB IC indexed 1 is inconsistent with the conjunction of revisions (indexing starts at 0)!*
9. conflict between $(\bigwedge EBIC)$ and contraction formulas: *The contraction indexed 6 is inconsistent with the conjunction of EB ICs (indexing starts at 0)!*
10. conflict between $(\bigwedge EBIC)$ and CB IC formulas: *The CB IC indexed 3 is inconsistent with the conjunction of EB ICs (indexing starts at 0)!*
11. conflicting pairs of CB IC formulas and contraction formulas: *The contraction indexed 2 is inconsistent with the CB IC indexed 0 (indexing starts at 0)!*

The aforementioned consistency checks correspond to the consistency checks on input in Algorithm $ComputeBCE$ from Section 3. Specifically, 1, 2, 3, and 6 correspond to the checks $(R \vdash \bot)$ and $(K \vdash \bot)$ in Line 1 of $ComputeBCE$; 5, 8, and 10 to the check ($R \cup \{\psi\} \vdash \bot$, for any $\psi \in IC_c$) in Line 2 of $ComputeBCE$; 4, 7, and 9 to the check ($R \cup \{\neg\phi\} \vdash \bot$, for any $\phi \in Con$) in Line 3 of $ComputeBCE$; lastly, 11 to the check ($\{\neg\phi\} \cup \{\psi\} \vdash \bot$, for any $\phi \in Con$ and any $\psi \in IC_c$) in Line 4 of $ComputeBCE$.

5 Experiments

It has been shown that skeptical revision and contraction in our approach are Π_2^P-hard problems [1]. In [10] is was shown how the approach could be encoded using quantified Boolean formulas (QBF). This allows us to compare COBA 2.0 with an implemented version of the approach using the quantified Boolean formula solver QUIP [11].

For comparing the implementations, we created knowledge bases and revision sentences made up of randomly generated 3-DNF formulas, and converted each to a QBF. We also devised an experimental prototype of COBA 2.0 which performs structural transformation (by replacing sub-formulas with new atoms) instead of the CNF conversion of formulas (for consistency checks). Experiments were then conducted on QUIP, and on both the stable version (the applet) and the experimental prototype of COBA 2.0.

Preliminary experimental results reveal that most of COBA 2.0's run-time is attributed to its structural or CNF conversion of formulas and to its consistency checks. The run-time of all three implementations shows an exponential growth rate. QUIP, however, is relatively faster than both versions of COBA 2.0. The experimental prototype seems to be more than two orders of magnitude faster than the stable version of COBA 2.0, and this observation suggests that structural transformation be done in lieu of CNF conversion in our future implementation.

6 Conclusion

We have presented COBA 2.0, an implementation of a consistency-based approach for belief change incorporating integrity constraints. Operators for belief revision and contraction incorporating integrity constraints are readily defined in a general framework that satisfies the majority of the AGM postulates, notably independence of syntactic representation As demonstrated by COBA 2.0, the framework is easily implementable, for the results of our operators are finite and vocabulary-restricted belief change can be performed instead. Examples of how COBA 2.0 computes belief change are detailed in Section 3.

Our preliminary experiments show that our stable version (the applet) still has much potential for improvement. To this end, we devised an experimental prototype (with structural transformation in lieu of CNF conversion) that seems to be more than two orders of magnitude faster than the stable version (with CNF conversion). Hence, we are optimistic that COBA 2.0 can be improved to achieve a similar run-time behaviour as the monolithic QUIP system.

To our knowledge, COBA 2.0 is the most general belief change system currently available, capable of computing arbitrary combinations of belief revision and contraction that (possibly) incorporate consistency-based and entailment-based integrity constraints. Moreover, COBA 2.0's general framework is easily extensible to consistency-based merging operators as detailed in [12], and currently we are refining our implementation so as to accommodate the merging of knowledge bases. The only comparable system is described in [13]. However, it is based on another approach to belief change, relying on stratified knowledge bases.

The applet, user manual, Java code, and Javadocs of COBA 2.0 are all accessible at [9].

References

1. Delgrande, J., Schaub, T.: A consistency-based approach for belief change. Artificial Intelligence 151, 1–41 (2003)
2. Alchourrón, C., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet functions for contraction and revision. Journal of Symbolic Logic 50, 510–530 (1985)
3. Fuhrmann, A.: Relevant Logics, Modal Logics, and Theory Change. PhD thesis, Australian National University, Australia (1988)
4. Katsuno, H., Mendelzon, A.: On the difference between updating a knowledge base and revising it. In: Gärdenfors, P. (ed.) Belief Revision, pp. 183–203. Cambridge University (1992)
5. Kowalski, R.: Logic for data description. In: Gallaire, H., Minker, J. (eds.) Logic and Data Bases, Plenum, pp. 77–103 (1978)
6. Sadri, F., Kowalski, R.: A theorem-proving approach to database integrity. In: Minker, J. (ed.) Foundations of Deductive Databases and Logic Programming, pp. 313–362. Morgan Kaufmann, San Francisco (1987)
7. Reiter, R.: Towards a logical reconstruction of relational database theory. In: Brodie, M., Mylopoulos, J., Schmidt, J. (eds.) On Conceptual Modelling, pp. 191–233. Springer, Heidelberg (1984)
8. A satisfiability library for java, `http://www.sat4j.org`
9. COBA 2.0., `http://www.cs.sfu.ca/~cl/software/COBA/coba2.html`

10. Delgrande, J., Schaub, T., Tompits, H., Woltran, S.: On computing belief change operations using quantified boolean formulas. Journal of Logic and Computation 14, 801–826 (2004)
11. Egly, U., Eiter, T., Tompits, H., Woltran, S.: Solving advanced reasoning tasks using quantified Boolean formulas. In: Proceedings of the AAAI National Conference on Artificial Intelligence, pp. 417–422 (2000)
12. Delgrande, J., Schaub, T.: Consistency-based approaches to merging knowledge bases. Journal of Applied Logics (to appear)
13. Benferhat, S., Kaci, S., Berre, D., Williams, M.A.: Weakening conflicting information for iterated revision and knowledge integration. Artificial Intelligence 153, 339–371 (2004)

An Algorithm for Computing Inconsistency Measurement by Paraconsistent Semantics *

Yue Ma[1,2], Guilin Qi[2], Pascal Hitzler[2], and Zuoquan Lin[1]

[1] Department of Information Science, Peking University, China
[2] Institute AIFB, Universität Karlsruhe, Germany
{mayue,lz}@is.pku.edu.cn, {yum,gqi,hitzler}@aifb.uni-karlsruhe.de

Abstract. Measuring inconsistency in knowledge bases has been recognized as an important problem in many research areas. Most of approaches proposed for measuring inconsistency are based on paraconsistent semantics. However, very few of them provide an algorithm for implementation. In this paper, we first give a four-valued semantics for first-order logic and then propose an approach for measuring the degree of inconsistency based on this four-valued semantics. After that, we propose an algorithm to compute the inconsistency degree by introducing a new semantics for first order logic, which is called $S[n]$-4 semantics.

1 Introduction

Measuring inconsistency in knowledge bases has been recognized as an important problem in many research areas, such as artificial intelligence [1,2,3,4,5], software engineering [6] and the Semantic Web [7]. There mainly exist two classes of inconsistency measures. The first class is defined by the number of formulas which are responsible for an inconsistency [8]. The second class considers propositions in the language which are affected by inconsistency [9,10,11,3,2]. The approaches belonging to the second class are often based on some paraconsistent semantics because we can still find paraconsistent models for inconsistent knowledge bases. The inconsistency degree considered in this paper belongs to the second class.

In [9], three compatible kinds of classifications for inconsistent theories are proposed, which actually provides three ways to define inconsistency measures for first-order logic based on paraconsistent semantics. The first approach is defined by the number of paraconsistent models. The underlying idea is that the less models, the more inconsistent the knowledge base is. The second approach is defined by the number of contradictions in a *preferred* paraconsistent model which has least contradictions, and considering the number of non-contradictions in a preferred model which has most non-contradictions. The third approach is defined by the number of atomic formulae which have conflicting assignments and by the number of all ground atomic formulae. Among

* We acknowledge support by the China Scholarship Council(http://www.csc.edu.cn/), by the German Federal Ministry of Education and Research (BMBF) under the SmartWeb project (grant 01 IMD01 B), by the EU under the IST project NeOn (IST-2006-027595, http://www.neon-project.org/), and by the Deutsche Forschungsgemeinschaft (DFG) in the ReaSem project.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 91–102, 2007.

these three approaches, the first one is global in the sense that all paraconsistent models are considered, while the latter two are local since they only consider the models with least inconsistencies or most consistencies. Later on, an approach for measuring inconsistency in first-order logic is given in [3], which is based on the third approach in [9].

Although there exist many approaches to measuring inconsistency in a knowledge base in a logical framework, very few of them provide efficient algorithms for implementation. In this paper, we first give a four-valued semantics for first-order logic and then propose an approach for measuring the degree of inconsistency based on this four-valued semantics. Our definition of inconsistency degree is similar to the approach given in [3]. The difference is that our approach is based on four-valued semantics and their approach is based on first-order quasi-classical semantics. After that, we propose an algorithm to compute the inconsistency degree by introducing a new semantics for first order logic, which is called $S[n]$-4 semantics.

This paper is organized as follows. In the next section, we introduce the four-valued semantics of first-order logic and its properties. In Section 3, we propose a definition of inconsistency degree of a first-order theory, and then, in Section 4, we give an algorithm to compute the inconsistency degree. Finally, we conclude the paper and discuss future work in Section 5.

2 Four-Valued First-Order Models

In order to measure the inconsistency degree of a first-order theory, in this section we define four-valued models for first-order theories. The inconsistency measurement studied in [2] is also by four-valued models. However, quantifiers and variables are not considered there — that is, only four-valued propositional models are being used. For first-order theories, an alternative semantic structure studied in [9] can be viewed as a three-valued semantics. Besides the definition of four-valued models, we also study how to reduce four-valued entailment to classical first-order entailment in this section, which serves as one of the important bases for our algorithm.

Given a set of predicate symbols $\mathcal{P}$ and a set of function symbols $\mathcal{F}$ (the set of 0-ary functions is a set of constant symbols, denoted $\mathcal{C}$), formulas are built up in the same way as in classical first-order logic from predicates, functions, a set of variables $\mathcal{V}$ and the set of logical symbols $\{\neg, \vee, \wedge, \forall, \exists, \rightarrow, \equiv\}$, where $\alpha \rightarrow \beta$ is the short form of $\neg\alpha \vee \beta$.

A first-order theory considered in this paper is a finite set of first-order formulae without free variables. In this paper, whenever we want to clarify the arity of a function or predicate, we may state the arity in parentheses following the function or predicate symbol, e.g. $f(n), P(n)$ means f, P are n-ary function and predicate, respectively. We also use t (possibly with subscripts) for terms, Greek lowercase symbols α, ϕ for formulas, and uppercase Γ for a first-order theory. The set of all predicates occurring in Γ is denoted as $\mathcal{P}(\Gamma)$. The cardinality of a set A is denoted by $|A|$.

The set of truth values for four-valued semantics [12,13] contains four elements: *true, false, unknown* (or *undefined*) and *both* (or *overdefined, contradictory*). We use the symbols t, f, N, B, respectively, for these truth values. The four truth values together with the ordering $\preceq$ defined below form a lattice $\mathsf{FOUR} = (\{t, f, B, N\}, \preceq)$:

$$f \preceq N \preceq t, f \preceq B \preceq t, N \text{ and } B \text{ are incomparable.}$$

The upper and lower bounds of two elements based on the ordering, and the operator $\neg$ on the lattice, are defined as follows:

- $N \wedge t = N, B \wedge t = B, N \wedge B = f$, and for any $x \in \mathsf{FOUR}$, $f \wedge x = f$;
- $f \vee N = N, f \vee B = B, N \vee B = t$, and for any $x \in \mathsf{FOUR}$, $t \vee x = t$;
- $\neg t = f, \neg f = t, \neg N = N, \neg B = B$, and for all $x \in \mathsf{FOUR}$, $\neg\neg x = x$.

Formally, a four-valued interpretation $\mathfrak{I}$ of a first-order theory is defined as follows.

Definition 1. *A four-valued interpretation $\mathfrak{I} = (\Delta^{\mathfrak{I}}, \cdot^{\mathfrak{I}})$ contains a non-empty domain $\Delta^{\mathfrak{I}}$ and a mapping $\cdot^{\mathfrak{I}}$ which assigns*

- *to each constant c an element of $\Delta^{\mathfrak{I}}$, written $c^{\mathfrak{I}}$;*
- *to each truth value symbol the symbol itself: $t^{\mathfrak{I}} = t, f^{\mathfrak{I}} = f, B^{\mathfrak{I}} = B, N^{\mathfrak{I}} = N$.*
- *to each n-ary function symbol $f(n)$ an n-ary function on $\Delta^{\mathfrak{I}}$, written $f^{\mathfrak{I}} : (\Delta^{\mathfrak{I}})^n \mapsto \Delta^{\mathfrak{I}}$, where $(\Delta^{\mathfrak{I}})^n = \overbrace{\Delta^{\mathfrak{I}} \times ... \times \Delta^{\mathfrak{I}}}^{n}$*
- *to each n-ary predication symbol $P(n)$ a pair of n-ary relations on $\Delta^{\mathfrak{I}}$, written $\langle P_+, P_- \rangle$, where $P_+, P_- \subseteq (\Delta^{\mathfrak{I}})^n$.*

Recall that a classical first-order interpretation maps each n-ary predicate to an n-ary relation on the domain. A four-valued interpretation assigns a pairwise n-ary relation $\langle P_+, P_- \rangle$ to each n-ary predicate P, where P_+ explicitly denotes the set of n-ary vectors which have the relation P under interpretation $\mathfrak{I}$ and P_- explicitly denotes the set of n-ary vectors which do not have the relation P under interpretation $\mathfrak{I}$. If a four-valued interpretation $\mathfrak{I}$ satisfies $P_+ \cup P_- = \Delta^{\mathfrak{I}}$ and $P_+ \cap P_- = \emptyset$, then it is a classical interpretation.

The definition of a state σ remains the same as in classical semantics of first-order logic, which is a mapping assigning to each variable occurring in $\mathcal{V}$ an element of the domain. Due to space limitation, we omit its formal definition as well as the definition of interpretation of terms based on states. We denote by $\sigma\{x \mapsto d\}$ the state obtained from σ by assigning d to x while leaving other assignments to other variables unchanged.

Given an interpretation $\mathfrak{I}$ and a state σ, the four-valued semantics of an atomic formula can be defined as follows.

Definition 2. *Assume $P(t_1, ..., t_n)$ is an n-ary predicate, where $t_1, ..., t_n$ are terms. $\mathfrak{I}$ is a four-valued interpretation and σ is a state. Then the truth value assignment to atomic predicates and equality is defined as follows:*

$$
\begin{aligned}
(x \equiv y)^{\mathfrak{I},\sigma} &= t, \ \textit{if and only if}\ x^\sigma = y^\sigma \\
(x \equiv y)^{\mathfrak{I},\sigma} &= f, \ \textit{if and only if}\ x^\sigma \neq y^\sigma \\
(P(t_1, ..., t_n))^{\mathfrak{I},\sigma} &= t, \ \textit{if and only if}\ (t_1^\sigma, ..., t_n^\sigma) \in P_+^{\mathfrak{I}} \ \textit{and}\ (t_1^\sigma, ..., t_n^\sigma) \notin P_-^{\mathfrak{I}} \\
(P(t_1, ..., t_n))^{\mathfrak{I},\sigma} &= f, \ \textit{if and only if}\ (t_1^\sigma, ..., t_n^\sigma) \notin P_+^{\mathfrak{I}} \ \textit{and}\ (t_1^\sigma, ..., t_n^\sigma) \in P_-^{\mathfrak{I}} \\
(P(t_1, ..., t_n))^{\mathfrak{I},\sigma} &= B, \ \textit{if and only if}\ (t_1^\sigma, ..., t_n^\sigma) \in P_+^{\mathfrak{I}} \ \textit{and}\ (t_1^\sigma, ..., t_n^\sigma) \in P_-^{\mathfrak{I}} \\
(P(t_1, ..., t_n))^{\mathfrak{I},\sigma} &= N, \ \textit{if and only if}\ (t_1^\sigma, ..., t_n^\sigma) \notin P_+^{\mathfrak{I}} \ \textit{and}\ (t_1^\sigma, ..., t_n^\sigma) \notin P_-^{\mathfrak{I}}
\end{aligned}
$$

where $\equiv$ is used for equality in first-order logic.

Note that the truth assignment to equality is classical in the sense that an equality can only obtain classical truth values t or f, while for common atomic predicates it may be valued among $\{t, f, B, N\}$. Based on the semantics of atomic predicates, the semantics of complex formulae can be defined deductively as follows:

Definition 3. *Let φ and ϕ be two first-order formulae, $\gamma(x_1, ..., x_n)$ be a formula containing n free variables, $\mathfrak{I}$ is a four-valued interpretation and σ is a state. Then,*

$$(\neg\varphi)^{\mathfrak{I},\sigma} = \neg(\varphi)^{\mathfrak{I},\sigma}; (\varphi \wedge \phi)^{\mathfrak{I},\sigma} = \varphi^{\mathfrak{I},\sigma} \wedge \phi^{\mathfrak{I},\sigma}; (\varphi \vee \phi)^{\mathfrak{I},\sigma} = \varphi^{\mathfrak{I},\sigma} \vee \phi^{\mathfrak{I},\sigma}$$

$$(\forall x_1, ..., x_n.\gamma(x_1, ..., x_n))^{\mathfrak{I},\sigma} = \bigwedge_{\sigma'=\sigma\{x_1\mapsto d_1,...,x_n\mapsto d_n\}} (\gamma(d_1, ..., d_n))^{\mathfrak{I},\sigma'}$$

$$(\exists x_1, ..., x_n.\gamma(x_1, ..., x_n))^{\mathfrak{I},\sigma} = \bigvee_{\sigma'=\sigma\{x_1\mapsto d_1,...,x_n\mapsto d_n\}} (\gamma(d_1, ..., d_n))^{\mathfrak{I},\sigma'}$$

Throughout the paper, we use the finite domain assumption such that the righthand of the last two equations above are finite conjunctions and disjunctions, respectively.

A four-valued interpretation $\mathfrak{I}$ is a 4-model of a first-order theory Γ if and only if for each formula $\alpha \in \Gamma$, $\alpha^{\mathfrak{I}} \in \{t, B\}$. A theory which has a 4-model is called 4-valued satisfiable. Four-valued entailment for first-order logic can be defined in a standard way as follows.

Definition 4. *Suppose Γ is a first-order theory and α is a first-order formula. Γ 4-valued entails α, written $\Gamma \models_4 \alpha$, if and only if every 4-model of Γ is a 4-model of α.*

Note that the four-valued interpretation of equality is the same as in classical first-order logic. So for all positive integers n, a four-valued interpretation $\mathfrak{I} = (\Delta^{\mathfrak{I}}, \cdot^{\mathfrak{I}})$ is a 4-model of formula $E_n = \exists x_1, ..., x_n. \bigwedge_{1\le i,j\le n}(x_i \not\equiv x_j) \wedge \forall y. \bigvee_{1\le i\le n}(y \equiv x_i)$ if and only if $|\Delta^{\mathfrak{I}}| = n$.

Proposition 1. *Given a first-order theory Γ without equality $\equiv$ and without boolean constants $\{t, f\}$, Γ always has 4-models of any domain size if UNA (the unique name assumption[1]) is not considered. If UNA is used, Γ always has 4-models whose sizes are equivalent to or larger than the number of constants in Γ.*

Example 1. *(Canonical example) $\Gamma = \{$Penguin$(tweety)$, Bird$(fred)$, $\forall x.$Bird$(x) \rightarrow$ Fly(x), $\forall x.$Penguin$(x) \rightarrow$ Bird(x), $\forall x.$Penguin$(x) \rightarrow \neg$Fly$(x)\}$. Obviously, Γ has no two-valued models. However, it has the following 4-model $\mathfrak{I} = (\Delta^{\mathfrak{I}}, \cdot^{\mathfrak{I}})$, where $\Delta^{\mathfrak{I}} = \{a, b\}$ and $\cdot^{\mathfrak{I}}$ is defined as tweety$^{\mathfrak{I}} = a$,fred$^{\mathfrak{I}} = b$, Fly$^{\mathfrak{I}}(a) = B$, Penguin$^{\mathfrak{I}}(a) =$ Bird$^{\mathfrak{I}}(a) =$ Bird$^{\mathfrak{I}}(b) =$ Fly$^{\mathfrak{I}}(b) = t$, Penguin$^{\mathfrak{I}}(b) = f$.*

According to Proposition 1, we restrict our measurement of the inconsistency degree to first-order theories which do not contain equality or $\{t, f\}$ in this paper.

Our four-valued semantics is an extension of classical semantics. Additionally, 4-valued entailment can be reduced to the classical entailment. The reduction in the propositional case is studied in [14]. We extend it to the first-order case.

[1] That is, if c and d are distinct constants, then $c^{\mathfrak{I}} \neq d^{\mathfrak{I}}$ for each interpretation $\mathfrak{I}$.

Theorem 1. *Let Γ be a first-order theory in negation normal form and ϕ be a formula. $\Gamma \models_4 \phi$ if and only if $\Theta(\Gamma) \vdash \Theta(\phi)$, where $\Theta(\cdot)$ is a function defined on a set of formulae as follows:*

- $\Theta(c) = c$, *if c is a constant.*
- $\Theta(\varphi) = \varphi$, *if φ is $x \equiv y$ or $x \not\equiv y$;*
- $\Theta(P(x_1, ..., x_n)) = P^+(x_1, ..., x_n)$, *where P^+ is a new atomic n-ary predicate;*
- $\Theta(\neg P(x_1, ..., x_n)) = P^-(x_1, ..., x_n)$, *where P^- is a new n-ary predicate;*
- $\Theta(\varphi_1(x_1, ..., x_n) \circ \varphi_2(y_1, ..., y_m)) = \Theta(\varphi_1(x_1, ..., x_n)) \circ \Theta(\varphi_2(y_1, ..., y_m))$, *where $\circ$ is $\wedge$ or $\vee$;*
- $\Theta(\varphi_1(x_1, ..., x_n) \rightarrow \varphi_2(y_1, ..., y_m)) = \Theta(\neg\varphi_1(x_1, ..., x_n)) \vee \Theta(\varphi_2(y_1, ..., y_m))$.
- $\Theta(Qx.\varphi) = Qx.\Theta(\varphi)$, *where Q is $\forall$ or $\exists$.*
- $\Theta(\Gamma) = \{\Theta(\varphi) \mid \varphi \in \Gamma\}$.

Example 2. *(Example 1 continued)*
$\Theta(\Gamma) = \{$*Penguin*$^+(tweety)$, *Bird*$^+(freg)$, $\forall x.$*Bird*$^-(x) \vee$ *Fly*$^+(x)$, $\forall x.$*Penguin*$^-(x) \vee$ *Bird*$^+(x)$, $\forall x.$*Penguin*$^-(x) \vee$ *Fly*$^-(x)\}$.

Example 3. *(Example 2 continued)*
Consider $\Gamma' = \Gamma \wedge$ *Fly*$(a_1) \wedge \neg$*Fly*$(a_1) \wedge E_n$ *and* $\varphi = \bigvee_{2 \le j \le n}($*Fly*$(a_j) \wedge \neg$*Fly*$(a_j)) \vee \bigvee_{1 \le j \le n}(($*Bird*$(a_j) \wedge \neg$*Bird*$(a_j)) \vee ($*Penguin*$(a_j) \wedge \neg$*Penguin*$(a_j)))$. *Obviously,* $\Theta(\Gamma') = \Theta(\Gamma) \wedge$ *Fly*$^+(a_1) \wedge$ *Fly*$^-(a_1)) \wedge E_n$ *and* $\Theta(\varphi) = \bigvee_{2 \le j \le n}($*Fly*$^+(a_j) \wedge$ *Fly*$^-(a_j)) \vee \bigvee_{1 \le j \le n}(($*Bird*$(a_j)^+ \wedge$ *Bird*$^-(a_j)) \vee ($*Penguin*$^+(a_j) \wedge$ *Penguin*$^-(a_j)))$. *According to Theorem 1, we know that* $\Gamma' \not\models_4 \varphi$ *because* $\Theta(\Gamma') \not\vdash \Theta(\varphi)$. *This example will be again used in Example 7.*

3 Inconsistency Measure by 4-Valued Semantics

To measure inconsistency of a theory, we consider only finite theory and only finite domains in this paper. This is reasonable for practical cases because only finite individuals can be represented or would be used.

Our approach to measuring inconsistency is based on the approach given in [3] which is defined by means of first-order quasi-classical models instead of four-valued models. The reason why we use 4-valued models is that the 4-valued semantics for the whole first-order language can be implemented by a linear reduction to the classical semantics. While for quasi-classical logic, this is only achieved restricted to propositional logic in CNF [15]. Due to space limitation, we omit all proofs. The underlying idea comes from [3].

Definition 5. *Let Γ be a first-order theory and $\mathfrak{I} = (\Delta^{\mathfrak{I}}, \cdot^{\mathfrak{I}})$ be a four-valued model of Γ. The* inconsistency degree *of Γ w.r.t. $\mathfrak{I}$, denoted* $Inc_{\mathfrak{I}}(\Gamma)$, *is a value in* $[0, 1]$ *calculated in the following way:*

$$Inc_{\mathfrak{I}}(\Gamma) = \frac{|ConflictTheo(\mathfrak{I}, \Gamma)|}{|GroundTheo(\mathfrak{I}, \Gamma)|}$$

where $GroundTheo(\mathfrak{I}, \Gamma) = \{P(d_1, ..., d_n) \mid d_1, ..., d_n \in \Delta^{\mathfrak{I}}, P(n) \in \mathcal{P}(\Gamma)\}$, *and* $ConflictTheo(\mathfrak{I}, \Gamma) = \{(P(d_1, ..., d_n))^{\mathfrak{I}} = B \mid d_1, ..., d_n \in \Delta^{\mathfrak{I}}, P(n) \in \mathcal{P}(\Gamma)\}$.

That is, the inconsistency degree of Γ w.r.t. $\mathfrak{I}$ is the ratio of the number of conflicting atomic sentences divided by the amount of all possible atomic sentences formed from atomic predicates occurring in Γ and individuals in the domain of $\mathfrak{I}$. It measures to what extent a given first-order theory Γ contains inconsistencies w.r.t. $\mathfrak{I}$.

Example 4. *(Example 1 continued)* $GroundTheo(\mathfrak{I}, \Gamma) = \{Bird(a), Penguin(a), Fly(a), Bird(b), Penguin(b), Fly(b)\}$, $ConflictTheo(\mathfrak{I}, \Gamma) = \{Fly(a)\}$. *So* $Inc_{\mathfrak{I}}(\Gamma) = \frac{1}{6}$.

Let's consider another 4-valued model $\mathfrak{I}'$ *of* Γ: $tweety^{\mathfrak{I}'} = a, fred^{\mathfrak{I}'} = b, Fly^{\mathfrak{I}'}(a) = Penguin^{\mathfrak{I}'}(a) = Bird^{\mathfrak{I}'}(a) = Bird^{\mathfrak{I}'}(b) = Fly^{\mathfrak{I}'}(b) = B, Penguin^{\mathfrak{I}'}(b) = f$. *Obviously,* $GroundTheo(\mathfrak{I}', \Gamma) = GroundTheo(\mathfrak{I}, \Gamma)$, $|GroundTheo(\mathfrak{I}', \Gamma)| = 5$, *and* $Inc_{\mathfrak{I}'}(\Gamma) = \frac{5}{6}$.

From this example, we can see that for any given first-order theory, its different 4-valued models might contain different percents of contradictions. According to this, we define a partial ordering on the set of its models as follows.

Definition 6. *(Model ordering w.r.t. inconsistency) Let* $\mathfrak{I}_1$ *and* $\mathfrak{I}_2$ *be two four-valued models of a first-order theory* Γ *such that* $|\Delta_1^{\mathfrak{I}}| = |\Delta_2^{\mathfrak{I}}|$. *We say that* $\mathfrak{I}_1$ *is less inconsistent than* $\mathfrak{I}_2$, *written* $\mathfrak{I}_1 \leq_{Incons} \mathfrak{I}_2$, *if and only if* $Inc_{\mathfrak{I}_1}(\Gamma) \leq Inc_{\mathfrak{I}_2}(\Gamma)$.

As usual, $\mathfrak{I}_1 <_{Incons} \mathfrak{I}_2$ denotes $\mathfrak{I}_1 \leq_{Incons} \mathfrak{I}_2$ and $\mathfrak{I}_2 \not\leq_{Incons} \mathfrak{I}_1$, and $\mathfrak{I}_1 \equiv_{Incons} \mathfrak{I}_2$ denotes $\mathfrak{I}_1 \leq_{Incons} \mathfrak{I}_2$ and $I_2 \leq_{Incons} \mathfrak{I}_1$. $\mathfrak{I}_1 \leq_{Incons} \mathfrak{I}_2$ means that $\mathfrak{I}_1$ is more consistent than $\mathfrak{I}_2$. The models of size n which are minimal w.r.t $\leq_{Incons}$ are called preferred models and they are formally defined as follows.

Definition 7. *Let* Γ *be a first-order theory,* $\mathcal{M}_4(\Gamma)$ *be the set of 4-models of* Γ, *and* $n(n \geq 1)$ *be a given cardinality.* Preferred models of size n w.r.t. $\leq_{Incons}$ *, written* $PreferModel_n(\Gamma)$, *are defined as follows:*

$$PreferModel_n(\Gamma) = \{\mathfrak{I} \mid |\Delta^{\mathfrak{I}}| = n; \forall \mathfrak{I}' \in \mathcal{M}_4(\Gamma), |\Delta^{\mathfrak{I}'}| = n \text{ implies } \mathfrak{I} \leq_{Incons} \mathfrak{I}'\}.$$

By Proposition 1 and Definition 7, it is not hard to see that given a first-order theory and an integer n, we can always find a preferred model if the unique name assumption is not used. Otherwise, with the unique name assumption, we only can find a preferred model provided n is not less than the number of constants appearing in the theory.

As a direct consequence of Definition 6 and Definition 7, the following corollary shows that for any two preferred four-valued models of a first-order theory with the same cardinality, the inconsistency degrees of the theory $w.r.t.$ them are equal.

Corollary 2. *Let* Γ *be a first-order theory and* $n(\geq 1)$ *be any given positive integer. Suppose* $\mathfrak{I}_1$ *and* $\mathfrak{I}_2$ *are two four-valued models of* Γ *such that* $|\Delta^{\mathfrak{I}_1}| = |\Delta^{\mathfrak{I}_2}| = n$, *and* $\{\mathfrak{I}_1, \mathfrak{I}_2\} \subseteq PreferModel_n(\Gamma)$. *Then* $Inc_{\mathfrak{I}_1}(\Gamma) = Inc_{\mathfrak{I}_2}(\Gamma)$.

Based on Corollary 2, the following definition of inconsistency degree of a first-order theory is well-defined.

Definition 8. *Given a first-order theory* Γ *and an arbitrary cardinality* $n(n \geq 1)$, *let* $\mathfrak{I}_n$ *be an arbitrary model in* $PreferModels_n(\Gamma)$. *The inconsistency degree of* Γ, *denoted by* $TheoInc(\Gamma)$, *is defined as* $\langle r_1, r_2, ..., r_n, ...\rangle$, *where* $r_n = *$ *if* $PreferModel_n(\Gamma) = \emptyset$, *and* $r_n = Inc_{\mathfrak{I}_n}(\Gamma)$ *otherwise. We use* $*$ *as a kind of null value.*

Following [3], we also use a sequence as the inconsistency degree of a first-order theory. This sequence can reflect the inconsistency information of the theory with respect to each finite size domain. For such sequences, the following property holds obviously.

Proposition 2. *Given an inconsistent first-order theory Γ, assume $|\mathcal{C}|$ is the number of constants of Γ and TheoInc$(\Gamma) = \langle r_1, r_2, ...\rangle$. Then for $i \geq |\mathcal{C}|$, $r_i \neq *$ and $r_i > 0$.*

This proposition shows that for any given first-order theory, its inconsistency measure cannot be a meaningless sequence (i.e., each element is the null value $*$) no matter whether UNA is used or not. Moreover, the non-zero values in the sequence start at least from the position which equals the number of constants in the first-order theory, and remains greater than zero in the latter positions of the sequence.

Example 5. *(Example 1 continued) If UNA is used, TheoInc$(\Gamma) = \langle *, \frac{1}{6}, ..., \frac{1}{3n}, ...\rangle$. If UNA is not used, TheoInc$(\Gamma) = \langle \frac{1}{3}, \frac{1}{6}, ..., \frac{1}{3n}, ...\rangle$. The 4-models which only assign Fly(tweety) to B are among the preferred models in both cases.*

4 Computational Aspects of Inconsistency Degree Sequences

A naive way to compute the inconsistency degree is to list all models to check which are the preferred models, and then compute the number of contradictions in these models. For a first-order theory, listing all models is not an easy and practical reasoning task.

In this section, we propose a practical way to compute the inconsistency degree by reducing the computation of the inconsistency degree to classical entailment, such that existing reasoners for first-order logic can be reused.

4.1 $S[n]$-4 Semantics

In this subsection, we define $S[n]$-4 semantics for first-order logic and show that $S[n]$-4 entailment can be reduced to classical entailment via four-valued entailment. We were inspired by [16]. $S[n]$-4 semantics will serve as the basis for our algorithm for computing the inconsistency degrees in Section 4.2.

Throughout this section, we assume that there is an underlying finite set of predicates $\mathcal{P}$ used for building all formulae and that $\mathcal{D}_n = \{a_1, ..., a_n\}$. The set of ground atomic formulae *Base*$(\mathcal{P}, \mathcal{D}_n)$ is defined as the set $\{P(a_{i_1}, ..., a_{i_m}) \mid P(m) \in \mathcal{P}, a_{i_1}, ..., a_{i_m} \in \mathcal{D}_n\}$.

Definition 9. *($S[n]$-4 Interpretation) Let $\mathcal{D}_n = \{a_1, ..., a_n\}$ be a domain of size n and S be any given subset of Base$(\mathcal{P}, \mathcal{D}_n)$. A 4-valued interpretation $\mathfrak{I}$ with domain $\mathcal{D}_n$ is called an $S[n]$-4 interpretation if and only if it satisfies the following condition:*

$$\phi^{\mathfrak{I}} = \begin{cases} B & \text{if } \phi \in \textit{Base}(\mathcal{P}, \mathcal{D}_n) \setminus S, \\ N \textit{ or } t \textit{ or } f & \text{if } \phi \in S \textit{ and } \{N, t, f\} \subseteq \textit{FOUR} \end{cases}$$

That is, $\mathfrak{I}$ is an $S[n]$-4 interpretation if and only if it is a 4-valued interpretation with domain of size n and assigns the contradictory truth value B to the ground atomic formulae not in S, and it maps non-contradictory truth values to ground atomic formulae in S.

Definition 10. *Let Γ be a first-order theory. An $S[n]$-4 interpretation $\mathfrak{I}$ is an $S[n]$-4 model of Γ if and only if it is a 4-model of Γ. A theory is $S[n]$-4 satisfiable if and only if it has an $S[n]$-4 model.*

Example 6. *Let $\mathcal{P} = \{p(x), q(x,y)\}$, $n = 2$, $\mathcal{D}_2 = \{a_1, a_2\}$. Then Base$(\mathcal{P}, \mathcal{D}_2) = \{p(a_1), p(a_2), q(a_1,a_1), q(a_2,a_2), q(a_1,a_2), q(a_2,a_1)\}$. Consider $\Gamma = \{\exists x.(p(x) \wedge \neg p(x)), \forall x \exists y.q(x,y)\}$.*

- *Let $S_1 = \{p(a_2), q(a_1,a_1), q(a_2,a_2), q(a_1,a_2), q(a_2,a_1)\}$. Γ is $S_1[2]$-4 satisfiable and has the following $S_1[2]$-4 model $\mathfrak{I}$: $p^{\mathfrak{I}}(a_1) = B$, and $\varphi^{\mathfrak{I}} = t$ for all $\varphi \in S_1$.*
- *Let $S_2 = \{p(a_1), p(a_2)\}$. Γ is $S_2[2]$-4 unsatisfiable since all $S_2[2]$-4 interpretations should map neither $p(a_1)$ nor $p(a_2)$ to B, so $\exists x.p(x) \wedge \neg p(x)$ cannot be satisfied.*

Theorem 3. *(Monotonicity) For any positive integer n, assume the two sets S and S' satisfying $S \subseteq S' \subseteq$ Base$(\mathcal{P}, \mathcal{D}_n)$. If a theory Γ is $S[n]$-4 unsatisfiable, then it is $S'[n]$-4 unsatisfiable.*

Proof. Assume that Γ is $S[n]$-4 unsatisfiable and that there exists an $S'[n]$-4 interpretation $\mathfrak{I}_{S'}$ satisfying Γ. We construct an $S[n]$-4 interpretation $\mathfrak{I}_S$ as follows.

$$\phi^{\mathfrak{I}_S} = \begin{cases} B & \text{if } \phi \in S' \setminus S, \\ \phi^{\mathfrak{I}_{S'}} & \text{otherwise.} \end{cases}$$

Obviously, $\mathfrak{I}_S$ is an $S[n]$-4 model of Γ, which is a contradiction. □

Definition 11. *($S[n]$-4 entailment) A sentence ϕ is $S[n]$-4 implied by a theory Γ, denoted $\Gamma \models^4_{S[n]} \phi$, if and only if every $S[n]$-4 model of Γ is an $S[n]$-4 model of ϕ.*

The relation between $S[n]$-4 satisfiability and $S[n]$-4 entailment is obvious.

Proposition 3. *Γ is $S[n]$-4 unsatisfiable if and only if $\Gamma \models^4_{S[n]} f$, where $f \in$ FOUR.*

The following theorem shows that $S[n]$-4 entailment can be reduced to 4-valued entailment in first-order logic.

Theorem 4. *For any $n \geq 1$ and $S \subseteq$ Base$(\mathcal{P}, \mathcal{D}_n)$, let $S = \{\alpha_1, ..., \alpha_m\}$ and $T =$ Base$(\mathcal{P}, \mathcal{D}_n) \setminus S = \{\beta_1, ..., \beta_k\}$, where $m + k = n$. Then the following claim holds:*

$$\Gamma \models^4_{S[n]} \varphi \text{ if and only if } \Gamma \wedge \bigwedge_{1 \leq i \leq k} (\beta_i \wedge \neg\beta_i) \wedge E_n \models_4 \varphi \vee \bigvee_{1 \leq j \leq m} (\alpha_j \wedge \neg\alpha_j),$$

where $E_n = \exists x_1, ..., x_n. \bigwedge_{1 \leq i,j \leq n} (x_i \not\equiv x_j) \wedge \forall y. \bigvee_{1 \leq i \leq n} (y \equiv x_i)$.

The right side of the claim is explained as follows: for each 4-model I of Γ, if I satisfies

1. it has an n-size domain (i.e., E_n is satisfied by I) and
2. it assigns truth value B to each element in Base$(\mathcal{P}, \mathcal{D}_n) \setminus S$ (i.e., the conjunction $\bigwedge_{1 \leq i \leq k} (\beta_i \wedge \neg\beta_i)$ is satisfied by I),

then I is not an $S[n]$-4 model of Γ if and only if it assigns B to at least one element in S (i.e., the disjunction $\bigvee_{1 \leq j \leq m} (\alpha_j \wedge \neg\alpha_j)$ is true under I). A formal proof is as follows.

Proof. Let $\Gamma' = \Gamma \wedge \bigwedge_{1\leq i\leq k}(\beta_i \wedge \neg\beta_i) \wedge E_n$ and let $\varphi' = \varphi \vee \bigvee_{1\leq j\leq m}(\alpha_j \wedge \neg\alpha_j)$.

($\Rightarrow$) For any 4-model M_4 of Γ', we show that M_4 satisfies φ'. First, from the assumption that M_4 satisfies Γ', we know $|\Delta^{M_4}| = n$ and $M_4(\beta_i) = B$ for $1 \leq i \leq k$. If there is j_0, $1 \leq j_0 \leq m$ such that $M_4(\alpha_{j_0}) = B$, then M_4 is a 4-valued model of φ'. Otherwise, if for each $1 \leq j \leq m$, $M_4(\alpha_j) \neq B$, then M_4 is an $S[n]$-4 model of Γ, so M_4 satisfies φ by hypothesis and therefore satisfies φ'.

($\Leftarrow$) For any $S[n]$-4 model M_S of Γ, we show that M_S satisfies φ. By definition of M_S, $|\Delta^{M_S}| = n$, $M_S(\beta_i) = B$ for $1 \leq i \leq k$, and $M_S(\alpha_j) \neq B$ for $1 \leq j \leq m$. So M_S is a 4-model of Γ' but does not satisfy $\bigvee_{1\leq j\leq m}(\alpha_j \wedge \neg\alpha_j)$. Then M_S satisfies φ by hypothesis and $\Gamma \models_{S\text{-}4} \varphi$. □

Corollary 5. *Let $S = \{\alpha_1, ..., \alpha_m\}$ and let $T = Base(\mathcal{P}, \mathcal{D}_n) \setminus S = \{\beta_1, ..., \beta_k\}$. Γ is $S[n]$-4 unsatisfiable if and only if*

$$\Theta(\Gamma \wedge \bigwedge_{1\leq i\leq k}(\beta_i \wedge \neg\beta_i)) \wedge E_n \vdash \bigvee_{1\leq j\leq m} \Theta((\alpha_j \wedge \neg\alpha_j)).$$

Proof. This corollary holds by replacing ϕ with f in Theorem 4 and then performing $\Theta(\cdot)$ according to Theorem 1 with the fact that $\Theta(E_n) = E_n$.

This theorem shows that $S[n]$-4 satisfiability can be reduced to classical entailment in first-order logic.

4.2 Algorithm for Computing the Inconsistency Degree

In this section, we first study how the inconsistency degree of an inconsistent theory Γ can be characterized by $S[n]$-4 satisfiability. Secondly, we give an algorithm to compute the inconsistency degree by invoking a classical reasoner.

Without loss of generality, throughout this section, we assume that the n-size ($n \geq 1$) domain of any 4-valued interpretation is $\mathcal{D}_n = \{a_1, ..., a_n\}$. Whenever we talk about $S[n]$-4 semantics used to compute the inconsistency degree of a first-order theory Γ, we always assume that the underlying finite set of predicates $\mathcal{P}$ is all the predicates occurring in Γ — that is, $\mathcal{P} = \mathcal{P}(\Gamma)$ and $Base(\mathcal{P}, \mathcal{D}_n) = GroundTheo(\mathcal{D}_n, \Gamma)$.

Theorem 6. *Let $TheoInc(\Gamma) = \langle r_1, ..., r_n, ...\rangle$. If $r_n \neq *$, the equation*

$$r_n = 1 - \frac{B_n}{|GroundTheo(\mathcal{D}_n, \Gamma)|} \tag{1}$$

holds, where $B_n = max\{|S| : S \subseteq GroundTheo(\mathcal{D}_n, \Gamma)$, so that Γ is $S[n]$-4 satisfiable$\}$.

Proof. Let $\mathfrak{I}_n$ be a preferred model and S be the set of atomic sentences all of which are not assigned the contradictory value B under $\mathfrak{I}_n$. Therefore, Γ is $S[n]$-4 satisfiable because $\mathfrak{I}_n$ is already an $S[n]$-4 model of Γ. For any subset $S' \subseteq GroundTheo(\mathcal{D}_n, \Gamma)$ such that $|S'| > |S|$, we claim that Γ is $S'[n]$-4 unsatisfiable. Otherwise suppose $\mathfrak{I}_{S'}$ is an $S'[n]$-4 model of Γ. Obviously, $\mathfrak{I}_{S'} <_{Incons} \mathfrak{I}_n$, since $|S'| > |S|$, contradicting the definition of $\mathfrak{I}_n$. Thus $B_n = |GroundTheo(\mathcal{D}_n, \Gamma)| - |ConflictTheo(\mathfrak{I}_n, \Gamma)|$. By Definition 5 and Definition 8, Equation 1 holds. □

Theorem 6 shows that the computation of r_n can be reduced to the problem of computing the maximal cardinality of S such that S is a subset of *GroundTheo*$(\mathcal{D}_n, \Gamma)$ and Γ is $S[n]$-4 satisfiable. We are now ready to give an algorithm to compute each element of the inconsistency degree sequence of a first-order theory Γ. The underlying idea is that we test $S[n]$-4 satisfiability for each subset S of *GroundTheo*$(\mathcal{D}_n, \Gamma)$ from size $|$*GroundTheo*$(\mathcal{D}_n, \Gamma)| - 1$ to 1. Whenever such subset has been found, the value of r_n is calculated by Equation 1 and the procedure ends.

Algorithm 1. Computing_Inconsistency_Degree(Γ, n)

Input: An inconsistent first-order theory Γ and a positive integer n
Output: r_n // *TheoInc*$(\Gamma) = \langle r_1, ..., r_n, ... \rangle$

1: $N \leftarrow$ the number of constants in Γ
2: **if** $n < N$ and UNA is used **then**
3: $r_n \leftarrow *$
4: **return** r_n
5: **end if**
6: $\mathcal{D}_n \leftarrow \{a_1, ..., a_n\}$,
7: $\Sigma \leftarrow$ *GroundTheo*$(\mathcal{D}_n, \Gamma)$ // see *GroundTheo*$(\mathcal{D}_n, \Gamma)$ in Definition 5
8: $r_n = 0$ // The initial value of r_n is set to 0
9: **for** $l \leftarrow |\Sigma| - 1$ to 1 **do**
10: $S \leftarrow PopSubset(\Sigma, l)$
// $PopSubset(\cdot, \cdot)$ is a procedure to return a subset of Σ with cardinality l. Once a subset is returned, it will not be selected again.
11: **while** $S \neq \emptyset$ **do**
12: **if** Γ is $S[n]$-4 satisfiable **then**
13: $r_n \leftarrow (1 - \frac{l}{|\Sigma|})$ **exit**
14: // $|S| = \max\{|S'| \mid S' \subseteq$ *GroundTheo*$(\mathcal{D}_n, \Gamma)$, Γ is $S'[n]$-4 satisfiable $\}$.
15: **else**
16: $S \leftarrow PopSubset(\Sigma, l)$
17: **end if**
18: **end while**
19: **if** $r_n \neq 0$ **then**
20: **exit** // The subset used to compute r_n has been found w.r.t. size l
21: **else**
22: $l \leftarrow l-1$ // We have to find a subset used to compute r_n w.r.t. a smaller cardinality.
23: **end if**
24: **end for**
25: **if** $l = 0$ **then**
26: $r_n = 1$
27: **end if**
28: **return** r_n

In Algorithm 1, if UNA is used and the input n is strictly less than the number of constants in Γ, then $r_n = *$ is returned (see line 2 to line 5). If it is not the case, the initialization process follows till line 8. From line 9 to line 27 we have the main steps of the algorithm to compute the inconsistency degree, where subsets of *GroundTheo*$(\mathcal{D}_n, \Gamma)$ are selected one by one according to a decreasing size ordering, so that whenever the

first subset S satisfying the condition in line 12, the inconsistency degree r_n is computed and the whole procedure ends. This is indeed the case because such S satisfies $B_n = |S| = \max\{|S'| \mid S' \subseteq \mathit{GroundTheo}(\mathcal{D}_n, \Gamma), \Gamma \text{ is } S'[n]\text{-4 satisfiable}\}$, where B_n is defined as in Theorem 6. Since Γ is inconsistent, it is no necessity to test $l = |\Sigma|$ in line 9. Furthermore, if no proper subset S of $\mathit{GroundTheo}(\mathcal{D}_n, \Gamma)$ can satisfy the condition in line 12, then this means that all sentences in $\mathit{GroundTheo}(\mathcal{D}_n, \Gamma)$ should be assigned B by preferred models, thus $r_n = 1$. This shows the correctness of this algorithm as well.

For line 12, the condition of $S[n]$-4 satisfiability can be decided by classical entailment of first-order logic according to Corollary 5, such that each r_n in the inconsistency degree sequence can be computed by invoking a classical reasoner. We give an example to illustrate Algorithm 1.

Example 7. *(Example 5 continued) We take the case that UNA is used and* $n \geq 2$. $\mathit{GroundTheo}(\mathcal{D}_n, \Gamma) = \{\mathit{Bird}(a_i), \mathit{Fly}(a_i), \mathit{Penguin}(a_i) \mid a_i \in \mathcal{D}_n\}$ *so that* $|\Sigma| = 3n$. *For* $l = |\Sigma| - 1 = 3n - 1$, *assume that following subset of* Σ *is selected:* $S = \mathit{GroundTheo}(\mathcal{D}_n, \Gamma) \setminus \{\mathit{Fly}(a_1)\}$. *We have that* Γ *is* $S[n]$*-4 satisfiable because of the result studied in Example 3. Then* $r_n = 1 - \frac{l}{3n} = \frac{1}{3n}$, *which equals the general representation of the inconsistency degree of* Γ *in Example 5.*

The computation of the inconsistency degree sequence $\langle r_1, ..., r_n, ...\rangle$ of a first-order theory Γ can be achieved using Algorithm 1. However, a practical problem is that the infinite style definition of $\mathit{TheoInc}(\Gamma)$ makes us unable to get the exact value of $\mathit{TheoInc}(\Gamma)$ in finite time. We can however set a termination condition in order to guarantee that an answer will be obtained. Suppose time (resource) is used up, a possible way is to use the already obtained partial sequences $\langle r_1, ..., r_n\rangle$ as an approximating value of $\mathit{TheoInc}(\Gamma)$.

From Theorem 6 and Corollary 5, the computation of each element of an inconsistency degree sequence includes at most $2^{|\Sigma|}$ times invoking a classical entailment, where $|\Sigma| \leq Kn^M$ for any $n \geq 1$ provided that the maximal arity of predicates in Γ is M and the number of predicates in Γ is K. The worst case occurs when all subsets of Σ have to be searched.

As to an optimization of the algorithm, the direct way is to properly design a procedure $PopSubset(\cdot, \cdot)$ such that the correct S which makes Γ $S[n]$-4 satisfiable can be found within as few steps as possible.

5 Conclusions and Future Work

In this paper, we have studied the computational aspects of calculating the inconsistency degree of a first-order theory. Theoretically, we have shown the process of encoding the calculation of the inconsistency degree as a first-order unsatisfiability decision problem via the $S[n]$-4 semantics proposed in this paper.

The semi-decidability of first-order logic makes Algorithm 1 semi-computes the inconsistency degree of first-order theory in the sense that we can be informed in finite time when Γ is $S[n]$-4 unsatisfiable for a chosen S; However if the correct subset of S such that Γ is $S[n]$-4 satisfiable is chosen, we actually cannot get the answer in finite

time in general cases. Therefore we also have to set a time termination condition for each computation of r_n, and when time is used up, Γ can be roughly considered to be $S[n]$-4 satisfiable and we can use this S to compute r_n.

Considering the semi-decidability problem, the study of implementing our algorithm on Description Logics which include a family of decidable fragments of first-order logic becomes meaningful [17].

In the future, we will study how to extend the underlying idea of our algorithm to compute other approaches to measuring inconsistency, such as the inconsistency degree defined in [3]. In order to provide inconsistency degree information for real applications, we will also consider approximating approaches to measuring inconsistency in future work.

References

1. Hunter, A.: Reasoning with contradictory information using quasi-classical logic. J. Log. Comput. 10, 677–703 (2000)
2. Hunter, A.: How to act on inconsistent news: Ignore, resolve, or reject. Data Knowl. Eng. 57, 221–239 (2006)
3. Grant, J., Hunter, A.: Measuring inconsistency in knowledgebases. J. Intell. Inf. Syst. 27(2), 159–184 (2006)
4. Hunter, A., Konieczny, S.: Shapley inconsistency values. In: Proc. of KR'06, pp. 249–259 (2006)
5. Konieczny, S., Lang, J., Marquis, P.: Quantifying information and contradiction in propositional logic through epistemic tests. In: Proc. of IJCAI'03, pp. 106–111 (2003)
6. Mu, K., Jin, Z., Lu, R., Liu, W.: Measuring inconsistency in requirements specifications. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 440–451. Springer, Heidelberg (2005)
7. Deng, X., Haarslev, V., Shiri, N.: Measuring inconsistencies in ontologies. In: ESWC, pp. 326–340 (2007)
8. Knight, K.: Measuring inconsistency. Journal of Philosophical Logic 31, 77–98 (2001)
9. Grant, J.: Classifications for inconsistent theories. Notre Dame J. Formal Logic 19, 435–444 (1978)
10. Hunter, A.: Measuring inconsistency in knowledge via quasi-classical models. In: Proc. of AAAI/IAAI, pp. 68–73 (2002)
11. Oller, C.A.: Measuring coherence using LP-models. J. Applied Logic 2, 451–455 (2004)
12. Belnap, N.D.: A useful four-valued logic. Modern uses of multiple-valued logics, 7–73 (1977)
13. Arieli, O., Avron, A.: The value of the four values. Artif. Intell. 102, 97–141 (1998)
14. Arieli, O., Denecker, M.: Reducing preferential paraconsistent reasoning to classical entailment. J. Log. Comput. 13, 557–580 (2003)
15. Marquis, P., Porquet, N.: Computational aspects of quasi-classical entailment. Journal of Applied Non-Classical Logics 11, 295–312 (2001)
16. Schaerf, M., Cadoli, M.: Tractable reasoning via approximation. Artif. Intell. 74, 249–310 (1995)
17. Ma, Y., Qi, G., Hitzler, P., Lin, Z.: Measuring inconsistency for description logics based on paraconsistent semantics. In: ECSQARU. LNCS (LNAI), vol. 4724, pp. 30–41. Springer, Heidelberg (2007)

How Dirty Is Your Relational Database? An Axiomatic Approach*

Maria Vanina Martinez[1], Andrea Pugliese[2], Gerardo I. Simari[1], V.S. Subrahmanian[1], and Henri Prade[3]

[1] University of Maryland, College Park, USA
{mvm,gisimari,vs}@cs.umd.edu
[2] University of Calabria, Rende, Italy
apugliese@deis.unical.it
[3] IRIT, Toulouse, France
henri.prade@irit.fr

Abstract. There has been a significant amount of interest in recent years on how to reason about inconsistent knowledge bases. However, with the exception of three papers by Lozinskii, Hunter and Konieczny and by Grant and Hunter, there has been almost no work on characterizing the degree of dirtiness of a database. One can conceive of many reasonable ways of characterizing how dirty a database is. Rather than choose one of many possible measures, we present a set of axioms that any dirtiness measure must satisfy. We then present several plausible candidate dirtiness measures from the literature (including those of Hunter-Konieczny and Grant-Hunter) and identify which of these satisfy our axioms and which do not. Moreover, we define a new dirtiness measure which satisfies all of our axioms.

1 Introduction

It is an open secret that most commercial databases are *dirty* and in fact, there is a wide range of companies (e.g. SAS, Ascential – previously known as Informix) that offer data cleaning services. However, to date, with the exception of some ground-breaking work [1,2,3], we are not aware of any work that attempts to actually characterize how dirty a database is, and thus there is no objective measure to assess whether an allegedly cleansed database is in fact significantly cleaner than the original.

In this paper, we focus on a more restricted scenario than [1,2,3]. We focus on inconsistency in just *relational* databases (i.e. tables of tuples) with associated functional dependencies [4] that form one of the most important types of integrity constraints used in databases. Intuitively, functional dependencies say that when certain attribute values are equal, then other attribute values must be equal as well. A good example of a functional dependency is one which says that in the same database, each person's salary is unique. We also assume the existence of a total order on attributes (i.e. columns) in the relational table that indicates how "reliable" those attributes are. Thus, in an employee

* Funded in part by grant N6133906C0149, ARO grant DAAD190310202, AFOSR grants FA95500610405 and FA95500510298, and NSF grant 0540216.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 103–114, 2007.

database, we may choose to believe that the social security number attribute is more reliable than the salary attribute. However, unlike Hunter-Konieczny's and Grant-Hunter's work which is primarily symbolic, our work is geared to inconsistency in numerical data – this is critical in real world databases as they almost always contain numeric data such as salary data, sales figures, shipping costs, etc.

We then propose a general set of three axioms that we believe *any* measure of database dirtiness must satisfy when a *single* functional dependency is present. We subsequently assess several dirtiness measures, both naive and from the past (including the Hunter-Konieczny and Grant-Hunter measures) to see how they comply with the axioms and show that they all fail to satisfy all the axioms. We finally define a new dirtiness measure which satisfies all the axioms. Subsequently, we show how any such single FD dirtiness measure can be used to build a dirtiness measure that handles multiple functional dependencies and we present a couple of example measures of this type.

2 Syntax and Notation

We assume the existence of a relational schema $R = (A_1, \ldots, A_n)$ [4] where the A_i's are attributes. Each attribute A_i has an associated domain, $dom(A_i)$. A *tuple* over R is a member of $dom(A_1) \times \cdots \times dom(A_n)$. A *database* DB is any finite set of tuples over R. In the rest of this paper, we assume R is arbitrary but fixed. For example, Fig. 1 shows a database over the schema (Name, Age, Height) with the obvious domains. Each row in the figure is a tuple. We assume the existence of a set of symbols called *tuple variables* that range over the tuples in DB. A *functional dependency* for database DB is any expression of the form $\forall t, t' \in DB$, $t.A_{i_1} = t'.A_{i_1} \wedge \ldots \wedge t.A_{i_t} = t'.A_{i_t} \Rightarrow t.A_{i_{t+1}} = t'.A_{i_{t+1}} \wedge \ldots \wedge t.A_{i_m} = t'.A_{i_m}$. For instance, in Fig. 1, $t.Name = t'.Name \Rightarrow t.Age = t'.Age$ is an example of a functional dependency saying that two tuples about the same person should agree on age.

Without loss of generality, we assume two functional dependencies cannot have the same antecedent. We assume that the attributes in a table are totally ordered by a *reliability ordering* $>_r$ (that can be derived, e.g., from inherent properties of attribute

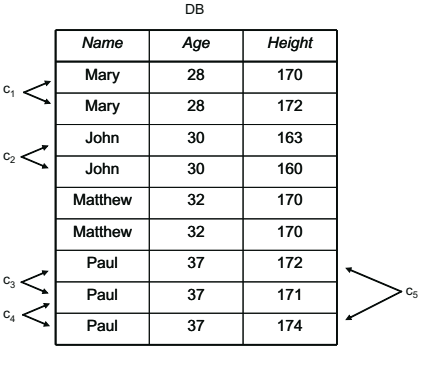


Fig. 1. An example database

domains or from historical error statistics). We write $A_i >_r A_j$ iff attribute A_i is more reliable than attribute A_j.

3 Culprits, Clusters, and Dirtiness Functions

Our notion of database dirtiness is based on the concepts of culprits and clusters. Culprits are just the duals of maximal consistent subsets which have been widely studied [1,2,3,5]. Clusters, on the other hand, do not seem to have been studied much in AI. Both of these parameters will be used in our axiomatic characterization of the dirtiness of a database.

Definition 1. *Let DB be a database and FD a set of functional dependencies. A* culprit *is a set $c \subseteq DB$ such that $c \cup FD$ is inconsistent and $\forall c' \subset c$, $c' \cup FD$ is consistent.*

Thus, culprits are minimal sets of database tuples that cause a functional dependency violation. Let $culprits(DB,\ FD)$ denote the set of culprits in DB w.r.t. FD.

Example 1. Consider a functional dependency fd stating that $\forall t, t' \in DB,\ t.Name = t'.Name \Rightarrow t.Age = t'.Age \wedge t.Height = t'.Height$. The relation in Fig. 1 has five culprits w.r.t. fd, denoted by $c_1,\ c_2,\ c_3,\ c_4,\ c_5$.

The following proposition states that the $culprits(DB, FD)$ function is monotonic w.r.t. DB.

Proposition 1. *If $DB' \subseteq DB$, then $culprits(DB',\ FD) \subseteq culprits(DB,\ FD)$.*

Definition 2. *Let DB be a database and FD a set of functional dependencies. Given two culprits $c, c' \in culprits(DB, FD)$, we say that c and c'* overlap*, denoted $c \triangle c'$, iff $c \cap c' \neq \emptyset$.*

Definition 3. *Let $\triangle^*$ be the reflexive transitive closure of relation $\triangle$. A* cluster *is a set $cl = \bigcup_{c \in e} c$ where e is an equivalence class of $\triangle^*$.*

We denote with $clusters(DB,\ FD)$ the set of all clusters in DB w.r.t. FD. We now present an example of overlapping culprits and clusters.

Example 2. In Fig. 1, the pairs of overlapping culprits in database DB are (c_1, c_1), (c_2, c_2), (c_3, c_3), (c_4, c_4), (c_5, c_5), (c_3, c_4), (c_3, c_5), (c_4, c_5), and all of the symmetric pairs. Therefore, the clusters in DB are the sets $cl_1 = \{(Mary, 28, 170), (Mary, 28, 172)\}$, $cl_2 = \{(John, 30, 163), (John, 30, 160)\}$, and $cl_3 = \{(Paul, 37, 172), (Paul, 37, 171), (Paul, 37, 174)\}$.

Clusters are important because they localize the inconsistencies. For instance, clusters cl_1, cl_2, cl_3 above tell us that there is something wrong with the *Mary, John* and *Paul* triples respectively.

We now define single-dependency and multiple-dependency dirtiness functions.

Definition 4. *A* single-dependency (resp. multiple-dependency) dirtiness function δ *takes a database instance DB, a functional dependency fd (resp. a finite set FD of functional dependencies), and a reliability ordering $>_r$ and returns as output a real number in the left-closed, right-open interval $[0, \infty)$.*

4 Axioms

Our first axiom on single-dependency dirtiness functions δ says that consistent databases have a dirtiness level of 0.

Axiom S1. If $culprits(DB, \{fd\}) = \emptyset$, then $\delta(DB, fd, >_r) = 0$.

Our second axiom is based on the statistical notions of standard deviation and variance (which is the square of s.d.), which have been used for decades by the statistics community as a measure of dirtiness in a data set, to define an axiom dirtiness functions should satisfy.

We first generalize the notion of variance to string attributes. Given a numeric attribute A, let $variance_A : 2^{dom(A)} \rightarrow \mathbb{R}^+$ be the variance of A. When $dom(A)$ is a set of strings, $variance_A$ builds on top of string similarity-evaluation function (e.g. edit distance, Hamming distance, Levenshtein distance). Given a set of strings S and a similarity-evaluation function $sim : string \times string \rightarrow \mathbb{R}^+$, let s_{min} be the first string appearing in S according to lexicographic order. The $variance_A(S)$ function returns the variance of the set $D = \{sim(s_{min}, s) \mid s \in S\}$.

From now on, the sequence of attributes in a functional dependency fd, ordered w.r.t. $>_r$, is denoted $\{A_{fd,1}, \ldots, A_{fd,m}\}$. Thus, $A_{fd,1}$ is the most reliable attribute in fd, $A_{fd,2}$ is the second most reliable attribute in fd, and so forth.

Definition 5. *Let fd be a functional dependency, and cl, cl' be two clusters. We say that $cl' \sqsubseteq^{fd}_{var} cl$, read "$cl'$ is less or equally varied than cl w.r.t. fd" iff $\exists j \in [1, m]$ s.t. $variance_{A_{fd,j}}(cl'.A_{fd,j}) \leq variance_{A_{fd,j}}(cl.A_{fd,j})$, and $\forall i < j, variance_{A_{fd,i}}(cl'.A_{fd,i}) = variance_{A_{fd,i}}(cl.A_{fd,i})$. We also say that $cl' \sqsubset^{fd}_{var} cl$, read "$cl'$ is less varied than cl w.r.t. fd" iff $\exists j \in [1, m]$ s.t. $variance_{A_{fd,j}}(cl'.A_{fd,j}) < variance_{A_{fd,j}}(cl.A_{fd,j})$, and $\forall i < j, variance_{A_{fd,i}}(cl'.A_{fd,i}) = variance_{A_{fd,i}}(cl.A_{fd,i})$.*

The above definition says that cl' is less or equally varied than cl w.r.t. fd iff as we examine the attribute in fd in decreasing order of reliability, the first attribute on which they have differing variances is one where cl' has a lower variance than cl.

Definition 6. *We say that DB'* is preferable to DB w.r.t. the dependency fd, *denoted $DB' \succ_{fd} DB$, iff there exists a function*

$$\alpha : clusters(DB', \{fd\}) \rightarrow clusters(DB, \{fd\})$$

such that $\forall cl' \in clusters(DB', \{fd\})$ it holds that:

- *$cl' \sqsubseteq^{fd}_{var} \alpha(cl')$;*
- *cl' and $\alpha(cl')$ agree on all attributes that appear in the body of fd;*

and at least one of the following conditions holds:

- *$\exists cl' \in clusters(DB', \{fd\})$ such that $cl' \sqsubset^{fd}_{var} \alpha(cl')$;*
- *$\exists cl \in clusters(DB, \{fd\})$ such that $\nexists cl' \in clusters(DB', \{fd\}), \alpha(cl') = cl$.*

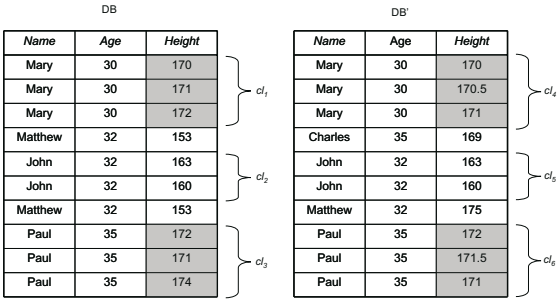


Fig. 2. According to Axiom S2, DB' has a lower dirtiness degree than DB

Intuitively, DB' is preferable to DB with respect to variance if there is a mapping between the clusters of DB' and the clusters of DB such that (*i*) each of the clusters in DB' shows less or equal variance than its image; (*ii*) either there exists a cluster in DB' having strictly less variance than its image in DB, or there exists a cluster in DB that does not belong to the codomain of the mapping.[1] This definition leads us directly to:

Axiom S2. If $DB' \succ_{fd} DB$, then $\delta(DB', fd, >_r) < \delta(DB, fd, >_r)$.

Example 3. Consider the databases in Fig. 2. Cluster cl_4 shows lower variance than cl_1; Clusters cl_2 and cl_5 are equal; Cluster cl_6 shows lower variance than cl_3. Therefore, Axiom S2 dictates that DB' has a lower dirtiness degree than DB.

We also consider a weaker variant of Axiom S2 called S2′:

Axiom S2′. If $DB' \succ_{fd} DB$ and $\forall cl' \in clusters(DB', \{fd\})\ cl' \subseteq \alpha(cl')$, then $\delta(DB', fd, >_r) < \delta(DB, fd, >_r)$.

The condition that $\forall cl' \in clusters(DB', \{fd\})\ cl' \subseteq \alpha(cl')$ is not satisfied by the databases in Fig. 2. Hence, Axiom S2′ does not impose any restrictions on δ. However, if (Mary, 30, 170.5) and (Paul, 35, 171.5) were not present in DB', then Axiom S2′ would instead require $\delta(DB', fd, >_r) < \delta(DB, fd, >_r)$.

5 Examples of Single-Dependency Dirtiness Functions

In this section, we present some single-dependency dirtiness functions.

5.1 Naive Culprits-Based Single-Dependency Dirtiness Functions

The following two simple dirtiness functions are based on culprits:

1. $|culprits(DB, \{fd\})|$

[1] It has been argued that when the values of disagreeing attributes are too far apart, they should simply be considered inconciliable [6]. In our case the objective is that of assessing the degree of dirtiness, so we still look at variances.

2. $\sum_{c \in culprits(DB,\{fd\})} |c|$

The first measure above just counts the number of culprits, the second sums up the number of tuples in each culprit.

Proposition 2. *The naive culprits-based dirtiness functions satisfy Axioms S1 and S2′.*

It is easy to see that these two measures, both of which seem reasonable at first sight, *do not satisfy Axiom S2*. To see why, consider the databases in Fig. 2. Here, we have $|culprits(DB, \{fd\})| = |culprits(DB', \{fd\})|$ and $\sum_{c \in culprits(DB,\{fd\})} |c| = \sum_{c \in culprits(DB',\{fd\})} |c|$, whereas Axiom S2 states that DB' should have a lower dirtiness degree.

5.2 Naive Cluster-Based Single-Dependency Dirtiness Functions

We now define two cluster-based dirtiness functions:

1. $|clusters(DB, \{fd\})|$
2. $\sum_{cl \in clusters(DB,\{fd\})} |cl|$

As in the case of the culprit based dirtiness functions, the first measure simply counts the number of clusters, while the second counts the sum of the number of tuples in each cluster.

Proposition 3. *Dirtiness function 1 above satisfies Axiom S1.*

It is easy to see that dirtiness function 1 above satisfies neither Axiom S2 nor S2′. To see why, consider the databases shown in Fig. 2. Here, $|clusters(DB, \{fd\})| = |clusters(DB', \{fd\})|$, whereas Axiom S2 states that DB' should have a lower dirtiness degree. Now consider DB' without tuples (Mary, 30, 170.5) and (Paul, 35, 171.5). We still have $|clusters(DB, \{fd\})| = |clusters(DB', \{fd\})|$, whereas Axiom S2′ states that DB' should have a lower dirtiness degree.

Proposition 4. *Dirtiness function 2 above satisfies Axioms S1 and S2′.*

Unfortunately, dirtiness function 2 above does not satisfy Axiom S2. To see why, consider the databases shown in Fig. 2. In this case, $\sum_{cl \in clusters(DB,\{fd\})} |cl| = \sum_{cl \in clusters(DB',\{fd\})} |cl|$, whereas Axiom S2 states that DB should have a higher dirtiness degree.

5.3 Functions Proposed in the Literature

In this section, we see how certain dirtiness functions proposed in the literature measure up w.r.t. the axioms we have proposed. The following function was proposed in [2]:

$$\frac{|culprits(DB, FD)|}{|DB \cup FD|}$$

This function looks at the ratio of the total number of culprits to the size of the database and functional dependencies.

Proposition 5. *The dirtiness function above satisfies Axiom S1.*

However, this dirtiness function does not satisfy either Axiom S2 nor S2′. The main reason is that this function does not look at the tuples inside a cluster. We consider the two axioms in turn:

(S2) Consider the databases shown in Fig. 3. We have $\frac{|culprits(DB,\{fd\})|}{|DB|+1} = \frac{3}{10}$ and $\frac{|culprits(DB',\{fd\})|}{|DB'|+1} = \frac{1}{3}$, thus contradicting Axiom S2 which states that DB' should have lower dirtiness than DB.

(S2′) The same example used for $S2$ shows that this function does not satisfy Axiom S2′.

The following dirtiness function was proposed in [3]:

$$\frac{|culprits(DB, FD)|}{|DB| + |ground(FD)|}$$

This function looks at the ratio of the number of culprits to the sum of the size of the database and the number of ground instances of the functional dependencies.

Proposition 6. *The dirtiness function above satisfies Axiom S1.*

However, this dirtiness function does not satisfy either Axiom S2 nor S2′ because of the fact that it does not examine clusters. We consider the two axioms in turn:

(S2) Consider the databases shown in Fig. 3. Suppose DB contains $x > 2k - 1$ more tuples which do not add to the number of inconsistencies that were already present. In this case we have $\frac{|culprits(DB',\{fd\})|}{|DB'|+k} = \frac{1}{3+k} > \frac{|culprits(DB,\{fd\})|}{|DB|+k}$, thus contradicting Axiom S2 which states that DB' should have lower dirtiness than DB.

(S2′) The same case considered for Axiom S2 shows that Axiom S2' is also contradicted.

The following function was proposed in [2,1]:

$$|DB| + |ground(FD)| - log_2|\bigcup_{\Delta \in MCS(DB \cup FD)} mod(\Delta)|$$

DB

Name	*Age*	*Height*
Mary	30	170
Mary	30	170
Matthew	25	153
Matthew	25	153
John	33	163
John	33	163
Paul	36	166
Paul	36	171
Paul	36	174

DB'

Name	*Age*	*Height*
Paul	36	171
Paul	36	174

Fig. 3. A case where the Grant-Hunter measure does not satisfy neither axiom Axiom S2 nor S2′

where $MCS(DB \cup FD)$ are the maximally consistent subsets of $DB \cup FD$ and $mod(\Delta)$ is the set of models of Δ. This function measures *cleanliness* with respect to functional dependencies, thus Axiom S1 is not applicable. If we take the negative of this function, the resulting dirtiness function does not satisfy Axioms S2 nor S2′. This can easily be seen by observing that adding or removing consistent tuples has a linear impact on the dirtiness measure while not changing the set of clusters.

5.4 A New Single-Dependency Dirtiness Function

Coming up with a single-dependency dirtiness function satisfying the axioms is a challenge. We now propose a new single-dependency dirtiness function δ_{var}. Let DB be a database, fd a functional dependency over DB, $\{A_{fd,1}, \ldots, A_{fd,m}\}$ the sequence of attributes in fd ordered w.r.t. $>_r$, and $variance_{max}(i)$, with $i \in [1, m]$, be the maximum possible variance for attribute $A_{fd,i}$. Let $B > 1$ be any integer. Then:

$$\delta_{var}(DB, fd, >_r) = \sum_{cl \in clusters(DB, \{fd\})} wtVar(cl, fd, >_r)$$

where

$$wtVar(cl, fd, >_r) = \sum_{i=1}^{m} B^{m-i} \cdot var'_{A_{fd,i}}(cl.A_{fd,i});$$

$$var'_{A_{fd,i}}(cl.A_{fd,i}) = (B-1) \cdot \frac{variance_{A_{fd,i}}(cl.A_{fd,i})}{variance_{max}(i)}.$$

Intuitively, we first compute the variance of each attribute $A_{fd,i}$ in each cluster cl, and normalize it to the range $[0, (B-1)]$ (this value is denoted as $var'_{A_{fd,i}}(cl.A_{fd,i})$). Then, for each cluster cl, we sum up the normalized variances of the attributes in fd, with exponentially decreasing weights (with base B) when going from the most reliable attribute to the less reliable one (this sum is denoted as $wtVar(cl, fd, >_r)$). The value of δ_{var} is finally computed as the sum of the $wtVar$'s of all the clusters. The following result says that δ_{var} satisfies all three axioms.

Theorem 1. *Function δ_{var} satisfies Axioms S1, S2, and S2′.*

The proof of the above theorem is rather involved, so we omit it for space reasons. Table 1 summarizes which dirtiness functions satisfy which axioms. Note that the only dirtiness function that satisfies all axioms is δ_{var}.

6 Combining Dirtiness w.r.t. Multiple Functional Dependencies

Most databases will have multiple functional dependencies. In the previous sections, we have looked at the situation where only one functional dependency is present. Combining dirtiness w.r.t. multiple functional dependencies can lead to anomalies.

Table 1. Single-dependency dirtiness functions

	S1	S2	S2′
$\lvert culprits(DB, \{fd\})\rvert$	✓	×	✓
$\sum_{c \in culprits(DB,\{fd\})} \lvert c\rvert$	✓	×	✓
$\lvert clusters(DB, \{fd\})\rvert$	✓	×	×
$\sum_{cl \in clusters(DB,\{fd\})} \lvert cl\rvert$	✓	×	✓
$\frac{\lvert culprits(DB,FD)\rvert}{\lvert DB \cup FD\rvert}$ [2]	✓	×	×
$\frac{\lvert culprits(DB,FD)\rvert}{\lvert DB\rvert + \lvert ground(FD)\rvert}$ [3]	✓	×	×
$\lvert DB\rvert + \lvert ground(FD)\rvert - log_2 \lvert \bigcup_{\Delta \in MCS(DB \cup FD)} mod(\Delta)\rvert$ [2,1]	n/a	×	×
δ_{var}	✓	✓	✓

Example 4. Consider the database in Fig. 4(a) and the following functional dependencies:

(fd_1) $\forall t, t' \in DB, t.Name = t'.Name \Rightarrow t.Age = t'.Age \wedge t.Salary = t'.Salary \wedge t.Position = t'.Position$
(fd_2) $\forall t, t' \in DB, t.Salary = t'.Salary \Rightarrow t.Position = t'.Position$

Here, $clusters(DB, \{fd_1\}) = \{cl_1, cl_2\}$, and $clusters(DB, \{fd_2\}) = \{cl_3\}$. If we look at the clusters with respect to both functional dependencies, i.e. if we consider $clusters(DB, \{fd_1, fd_2\})$, then we obtain the set of all five tuples.

The following definition specifies what it means for a database to be "clearly cleaner" than another database.

Definition 7. *Given a single-dependency dirtiness function δ, we say that $DB' \gtrsim_{FD} DB$, read "DB' is clearly cleaner than DB with respect to the set of dependencies FD", iff $\forall fd \in FD$, $\delta(DB', fd, >_r) \leq \delta(DB, fd, >_r)$.*

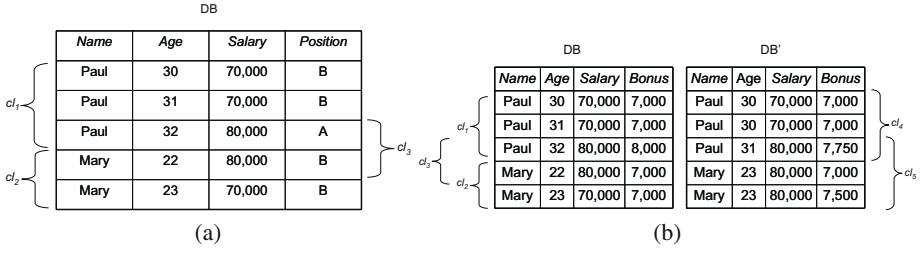


Fig. 4. (a) A case where a single cluster may comprise tuples violating different functional dependencies; (b) According to Axiom M1, DB' is clearly cleaner than DB

Suppose τ is a function that measures the dirtiness of a database DB based on a reliability ordering $>_r$ and a set of functional dependencies, and suppose τ uses a single-dependency dirtiness function δ to measure dirtiness in a database w.r.t. a single functional dependency. Then we hypothesize that τ needs to satisfy the following axiom.

Axiom M1. If $DB' \gtrsim_{FD} DB$, then $\tau(DB', FD, >_r) \leq \tau(DB, FD, >_r)$.

This axiom merely says that if DB' is clearly cleaner than DB, then τ must assign a lower (or equal) level of dirtiness to DB'.

Example 5. Consider the databases in Fig. 4(b) and the following functional dependencies:

(fd_1) $\forall t, t' \in DB, t.Name = t'.Name \Rightarrow t.Age = t'.Age \wedge t.Salary = t'.Salary$
(fd_2) $\forall t, t' \in DB, t.Salary = t'.Salary \Rightarrow t.Bonus = t'.Bonus$

Here, $clusters(DB, \{fd_1\}) = \{cl_1, cl_2\}$, $clusters(DB, \{fd_2\}) = \{cl_3\}$, $clusters(DB', \{fd_1\})$=$\{cl_4\}$, and $clusters(DB', \{fd_2\})$=$\{cl_5\}$. We can clearly see that $\delta(DB', fd_1, >_r) < \delta(DB, fd_1, >_r)$ and $\delta(DB', fd_2, >_r) < \delta(DB, fd_2, >_r)$. Therefore, Axiom M1 dictates that $\tau(DB', FD, >_r) \leq \tau(DB, FD, >_r)$.

We now propose two dirtiness functions that support multiple functional dependencies, both of which build on top of a single-dependency dirtiness function. Thus, even though our axioms on multiple-dependency dirtiness functions are weak (because there is only one axiom), things are actually more constrained than might be immediately apparent because they are required to build on top of a single-dependency dirtiness function.

The first function we propose makes the conservative choice of taking the maximum among the values returned by the single-dependency function.

Definition 8 (Pessimistic multiple-dependency dirtiness function). *Let DB be a database, FD a set of functional dependencies over DB, $>_r$ an ordering of the attributes of DB, and δ a single-dependency dirtiness function: We define function τ_{max} as*

$$\tau_{max}(DB, FD, >_r) = max_{fd \in FD} \delta(DB, fd, >_r)$$

It is immediate to see that this multiple-dependency dirtiness function satisfies Axiom M1.

Proposition 7. *Function τ_{max} satisfies Axiom M1.*

The second dirtiness function takes into account the fact that some functional dependencies might be more important than others, so violations of less important dependencies should contribute less to dirtiness.

Definition 9 (Preference-based multiple-dependency dirtiness function). *Let DB be a database, FD a set of functional dependencies over DB, $>_r$ an ordering of the attributes of DB, δ a single-dependency dirtiness function, and $weight : FD \rightarrow \mathbb{N}^+$: We define function τ_{wt} as*

$$\tau_{wt}(DB, FD, >_r) = \frac{\sum_{fd \in FD} weight(fd) \cdot \delta(DB, fd, >_r)}{\sum_{fd \in FD} weight(fd)}$$

The following straightforward result says that τ_{wt} also satisfies Axiom M1.

Proposition 8. *Function τ_{wt} satisfies Axiom M1.*

A special case of τ_{wt} takes the average of the dirtiness values returned by the single-dependency function:

$$\tau_{avg}(DB, FD, >_r) = \frac{\sum_{fd \in FD} \delta(DB, fd, >_r)}{|FD|}$$

obtained by setting in τ_{wt}, $\forall fd \in FD,\ weight(fd) = k$ for any fixed $k \in \mathbb{N}^+$.

7 Related Work and Conclusions

There has been a tremendous amount of work in inconsistency management since the 60s and 70s when paraconsistent logics where introduced [7] and logics of inconsistency [8,9] were developed. Subsequently, frameworks such as default logic [10], maximal consistent subsets [5] and inheritance networks [11] and others were used to generate multiple plausible consistent scenarios (often called "extensions"), and methods to draw inferences were developed that looked at truth in all (or some) extensions. Argumentation methods [12] were used to reason about how certain arguments defeated others. Methods to clean data and/or provide consistent query answers in the presence of inconsistent data are also quite common [13,14,15,16]. For instance, [15] addresses the basic concepts and results of the area of consistent query answering (in the standard model-theoretic sense). They consider universal and binary integrity constraints, denial constraints, functional dependencies, and referential integrity constraints. [16] presents a cost-based framework that allows finding "good" repairs for databases that exhibit inconsistencies in the form of violations to either functional or inclusion dependencies. They propose heuristic approaches to constructing repairs based on equivalence classes of attribute values; the algorithms presented are based on greedy selection of least repair cost, and a number of performance optimizations are also explored.

However, we are aware of very few works on measuring the degree of inconsistency in a database. All three methods deal with culprits only or with maximal consistent subsets [1,2,3]. We believe we have made two important conceptual contributions in this paper. First, we draw attention to the notion of a *cluster* and explain that clusters are very important in measuring cleanliness of the database. Second, we have drawn attention to the fact that well known statistical measures for measuring variation in a dataset (such as standard deviation and variance) have a role to play in measuring the dirtiness of a database. Based on these two ideas, we have developed single-dependency axioms that we believe a dirtiness measure should satisfy when one functional dependency is considered in isolation. We subsequently look at some obvious dirtiness measures based on culprits and clusters, as well as past work, and show that these methods do not satisfy our axioms. We then develop our own dirtiness measure that satisfies these

axioms. Subsequently, we propose a single axiom for dirtiness functions that handle multiple functional dependencies – however, such dirtiness functions are supposed to be built on top of a dirtiness function for single dependencies. We present a couple of alternative dirtiness functions that satisfy this axiom.

Future work will focus on the development of other multiple-dependency dirtiness functions and experimental evaluations of how these dirtiness functions work in practice in terms of computational overhead they impose. Moreover, we plan to build "cleaning" operators that provably reduce dirtiness.

References

1. Lozinskii, E.L.: Resolving contradictions: A plausible semantics for inconsistent systems. J. of Automated Reasoning 12(1), 1–31 (1994)
2. Hunter, A., Konieczny, S.: Approaches to measuring inconsistent information. In: Inconsistency Tolerance, pp. 191–236 (2005)
3. Grant, J., Hunter, A.: Measuring inconsistency in knowledgebases. J. Intell. Inf. Syst. 27(2), 159–184 (2006)
4. Ullman, J.: Principles of Data Base and Knowledge Base Systems. Addison-Wesley, Reading (1989)
5. Baral, C., Kraus, S., Minker, J., Subrahmanian, V.: Combining knowledge bases consisting of first order theories. Computational Intelligence (1992)
6. Bosc, P., Dubois, D., Pivert, O., Prade, H., de Calmes, M.: Fuzzy summarization of data using fuzzy cardinalities. In: International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems, pp. 1553–1559 (2002)
7. da Costa, N.: On the theory of inconsistent formal systems. Notre Dame Journal of Formal Logic 15(4), 497–510 (1974)
8. Belnap, N.: A useful four valued logic. In: Epstein, G., Dunn, M. (eds.) Modern Uses of Many Valued Logic, pp. 8–37 (1977)
9. Grant, J.: Classifications for inconsistent theories. Notre Dame Journal of Formal Logic 19(3), 435–444 (1978)
10. Reiter, R.: A logic for default reasoning. Artificial Intelligence (1980)
11. Touretzky, D.: The mathematics of inheritance systems. Morgan Kaufmann, San Francisco (1986)
12. Amgoud, L., Cayrol, C.: A reasoning model based on the production of acceptable arguments. Annals of Mathematics and Artificial Intelligence 34(1-3), 197–215 (2002)
13. Jermyn, P., Dixon, M., Read, B.J.: Preparing clean views of data for data mining. In: ERCIM Workshop on Database Research, pp. 1–15 (1999)
14. Schallehn, E., Sattler, K.: Using Similarity-based Operations for Resolving Data-level Conflicts. In: James, A., Younas, M., Lings, B. (eds.) British National Conference on Databases. LNCS, vol. 2712, pp. 172–189. Springer, Heidelberg (2003)
15. Chomicki, J.: Consistent query answering: Five easy pieces. In: ICDT, pp. 1–17 (2007)
16. Bohannon, P., Fan, W., Flaster, M., Rastogi, R.: A cost-based model and effective heuristic for repairing constraints by value modification. In: SIGMOD, pp. 143–154 (2005)

A Top-Down Query Answering Procedure for Normal Logic Programs Under the Any-World Assumption

Umberto Straccia

ISTI - CNR
Via G. Moruzzi, 1
56124 Pisa ITALY
straccia@isti.cnr.it

Abstract. The Any-World Assumption (AWA) has been introduced for normal logic programs as a generalization of the well-known notions of Closed World Assumption (CWA) and the Open World Assumption (OWA). The AWA allows *any* assignment (i.e., interpretation), over a *truth space* (bilattice), to be a default assumption and, thus, the CWA and OWA are just special cases. To answer queries, we provide a novel and simple top-down procedure.

1 Introduction

The *Any-World Assumption* (AWA) for normal logic programs [16] is a generalization of the notions of the Closed World Assumption (CWA) (which asserts that by default the truth of an atom is *false*) and the Open World Assumption (OWA) (which asserts that the truth of the atoms is supposed to be *unknown* by default). Essentially, the AWA allows *any* interpretation over a *truth space* to be a default assumption. The truth spaces considered are so-called *bilattices* [13] and the semantics generalizes the notions of Kripke-Kleene, well-founded and stable model semantics [10,11,26].

The AWA has many applications (see [16]), among which: (i) *Extended Logic Programs* (ELPs) (e.g., [2,3,12]); (ii) many-valued logic programming with non-monotone negation (e.g., [5,23]); (iii) *paraconsistency* (e.g., [1,3,4]) and (iv)representation of default rules by relying on the so-called *abnormality theory* [19].

In [16] a declarative and a fixed-point characterization for the AWA is presented. As a consequence, in order to answer queries we have to compute the intended model I of a logic program $\mathcal{P}$ by a bottom-up fixed-point computation and then answer with $I(A)$. [24] provides a top-down query answering procedure. However, it requires the grounding of the logic program. Furthermore, queries are ground atoms only. This approach is clearly not satisfactory in case we are looking for *all* answers to a query atom of the form $q(\mathbf{x})$. Indeed, the size of the grounded instance of a logic program as well as the number of query instances $q(\mathbf{c})$ to query may be large and generally exponential with respect to the size of the non-ground expressions.

In this paper we further improve the query answering procedure related to the AWA. We present a simple, yet general top-down query answering procedure, which focuses on computing *all* answers of a query. This is important as it is quite natural that a user would like the answers $\mathbf{c}$ to a query $q(\mathbf{x})$ be *ranked* according to the degree of

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 115–127, 2007.

$q(\mathbf{c})$. Essentially, the basic idea of our procedure is to collect, during the computation, all correct answers incrementally together in a similar way as it is done for classical Datalog [25]. Hence, for instance, we do not rely on any notion of *atom unification*, but rather iteratively access relational tables using relational algebra. Besides being the procedure novel for the AWA, we get for free a novel top-down query procedure for many-valued normal logic programs. This is the first time the issue of computing all answers has been addressed for many-valued normal logic programs under the OWA, CWA or more generally under the AWA in a many-valued semantics setting.

We proceed as follows. In the next two sections we recall concisely the AWA (we refer the interested reader to [16]). Then we present our top-down query procedure.

2 Preliminaries

Bilattice. The truth spaces we consider are *bilattices* [13]. Bilattices play an important role in (especially in theoretical aspects of) logic programming, and in knowledge representation in general, allowing to develop unifying semantical frameworks [10]. A *bilattice* [13,10] is a structure $\mathcal{B} = \langle B, \preceq_t, \preceq_k \rangle$ where B is a non-empty set and $\preceq_t$ (the *truth order*) and $\preceq_k$ (the *knowledge order*) are both partial orderings giving B the structure of a *complete lattice*. *Meet (or greatest lower bound)* and *join (or least upper bound)* under $\preceq_t$ are denoted $\wedge$ and $\vee$, while *meet and join under* $\preceq_k$ are denoted $\otimes$ and $\oplus$. *Top and bottom under* $\preceq_t$ are denoted `t` and `f`, and *top and bottom under* $\preceq_k$ are denoted $\top$ and $\bot$, respectively. We assume that each bilattice has a *negation*, i.e., an operator $\neg$ that reverses the $\preceq_t$ ordering, leaves unchanged the $\preceq_k$ ordering, and verifies $\neg\neg x = x$ [1]. We also provide a family $\mathcal{F}$ of $\preceq_k$ and $\preceq_t$-monotone n-ary functions over B to manipulate truth values. Furthermore, we assume that bilattices are *infinitary distributive bilattices* in which all distributive laws connecting $\wedge, \vee, \otimes$ and $\oplus$ hold. Finally, we also assume that every bilattice satisfies the *infinitary interlacing conditions*, i.e., each of the lattice operations $\wedge, \vee, \otimes$ and $\oplus$ is monotone w.r.t. both orderings (e.g., $x \preceq_t y$ and $x' \preceq_t y'$ implies $x \otimes x' \preceq_t y \otimes y'$).

Generalized Logic Programs. We extend logic programs where *computable functions* $f \in \mathcal{F}$ are allowed to manipulate truth values (see [23,24]). [2] That is, we allow any $f \in \mathcal{F}$ to appear in the body of a rule to be used to combine the truth of the atoms appearing in the body. The language is sufficiently expressive to accommodate almost all frameworks on many-valued logic programming with or without negation [23].

A *term*, t, is either a variable or a constant symbol. An *atom*, A, is an expression of the form $p(t_1, \ldots, t_n)$, where p is an n-ary predicate symbol and all t_i are terms. A literal, L, is of the form A or $\neg A$, where A is an atom. A *formula*, φ, is an expression built up from the literals, the truth values $b \in B$ of the bilattice and the functions $f \in \mathcal{F}$. Note that the members of the bilattice (i.e., truth values) may appear in a formula, as well as functions $f \in \mathcal{F}$. A *rule* is of the form $A \leftarrow \varphi$ where A is an atom

[1] The dual operation to negation is *conflation* i.e., an operator $\sim$ that reverses the $\preceq_k$ ordering, leaves unchanged the $\preceq_t$ ordering, and $\sim\sim x = x$. We do not deal with conflation in this paper.

[2] With computable we mean that for any input, the value of f can be determined in finite time.

Table 1. Models, Kripke-Kleene, well-founded and H-founded models of $\mathcal{P}$

	I_i							
	$q(a)$	$q(b)$	$r(a)$	$r(b)$	$p(a)$	$p(b)$	$KK(\mathcal{P})$	$WF(\mathcal{P})$
I_1	$\bot$	t	t	f	$\bot$	$\bot$	•	
I_2	f	t	t	f	f	f		•
I_3	t	t	t	f	t	f		

	H_i					
	$q(a)$	$q(b)$	$r(a)$	$r(b)$	$p(a)$	$p(b)$
H_1	f	f	f	f	f	f
H_2	f	f	f	f	f	f
H_3	t	f	f	f	t	f

	$s_{\mathcal{P}}^{H_i}(I_i)$						$U_{\mathcal{P}}(I_i)$
	$q(a)$	$q(b)$	$r(a)$	$r(b)$	$p(a)$	$p(b)$	
I_1	f	$\bot$	$\bot$	f	f	f	$\{q(a), r(b), p(a), p(b)\}$
I_2	f	$\bot$	$\bot$	f	f	f	$\{q(a), r(b), p(a), p(b)\}$
I_3	$\bot$	$\bot$	$\bot$	f	t	f	–

and φ is a formula. For instance, $p \leftarrow max(0, q + r - 1)$ is a rule dictating that p is at least as true as the conjunction of q and r with respect to the Lukasiewicz t-norm $x \wedge y = \max(0, x + y - 1)$. A *generalized normal logic program*, or simply *logic program*, $\mathcal{P}$, is a finite set of rules.

The notions of *Herbrand universe* $H_{\mathcal{P}}$ of $\mathcal{P}$ and *Herbrand base* (as the set of all ground atoms) $B_{\mathcal{P}}$ of $\mathcal{P}$ are as usual. Additionally, given $\mathcal{P}$, the generalized normal logic program $\mathcal{P}^*$ is constructed as follows: (i) set $\mathcal{P}^*$ to the set of all ground instantiations of rules in $\mathcal{P}$; (ii) replace several rules in $\mathcal{P}^*$ having same head, $A \leftarrow \varphi_1$, $A \leftarrow \varphi_2, \ldots$ with $A \leftarrow \varphi_1 \vee \varphi_2 \vee \ldots$ (recall that $\vee$ is the join operator of the bilattice); and (iii) if an atom A is not head of any rule in $\mathcal{P}^*$, then add the rule $A \leftarrow \mathtt{f}$ to $\mathcal{P}^*$ (it is a standard practice in logic programming to consider such atoms as *false*). This already acts as a kind of default assumption on non-derivable facts. We will change this point once we allow any default value as assumption later one. Note that in $\mathcal{P}^*$, each atom appears in the head of *exactly one* rule and that $\mathcal{P}^*$ is *finite*.

We next recall the usual semantics of logic programs over bilattices (cf. [16]). For ease, we will rely on the following simple example.

Example 1. Consider the logic program $\mathcal{P}$ with the following rules. $q(x) \leftarrow q(x) \vee \neg r(x)$, $p(x) \leftarrow p(x)$, $r(a) \leftarrow \mathtt{t}$, and $r(b) \leftarrow \mathtt{f}$. In Table 1 we report three models I_i of $\mathcal{P}$, the Kripke-Kleene and the well-founded model of $\mathcal{P}$ marked by bullets. The other tables will be discussed later on.

Interpretations. An *interpretation* I on the bilattice $\mathcal{B} = \langle B, \preceq_t, \preceq_k \rangle$ is a mapping from atoms to members of B. I is extended from atoms to formulae in the usual way: (i) for $b \in B$, $I(b) = b$; (ii) for formulae φ and φ', $I(\varphi \wedge \varphi') = I(\varphi) \wedge I(\varphi')$, and similarly for $\vee, \otimes, \oplus$ and $\neg$; and (iii) for formulae $f(A)$, $I(f(A)) = f(I(A))$, and similarly for n-ary functions. $\preceq_t, \preceq_k$ are extended from B to the set $\mathcal{I}(B)$ of all interpretations point-wise: (i) $I_1 \preceq_t I_2$ iff $I_1(A) \preceq_t I_2(A)$, for every ground atom A; and (ii) $I_1 \preceq_k I_2$ iff $I_1(A) \preceq_k I_2(A)$, for every ground atom A. With $I_{\mathtt{f}}$ and $I_{\bot}$ we denote the bottom interpretations under $\preceq_t$ and $\preceq_k$ respectively (they map any atom into f and $\bot$, respectively). $\langle \mathcal{I}(B), \preceq_t, \preceq_k \rangle$ is a bilattice as well.

Models. I is a *model* of $\mathcal{P}$, denoted $I \models \mathcal{P}$, iff for all $A \leftarrow \varphi \in \mathcal{P}^*$, $I(A) = I(\varphi)$. Note that usually a model has to satisfy $I(\varphi) \preceq_t I(A)$ only, i.e., $A \leftarrow \varphi \in \mathcal{P}^*$ specifies the necessary condition on A, "A is at least as true as φ". But, as $A \leftarrow$

$\varphi \in \mathcal{P}^*$ is the unique rule with head A, the constraint becomes also sufficient (see e.g., [10,16,17]). Among all the models, two models play a special role: namely the *Kripke-Kleene model* ($KK_{\mathcal{P}}$), which is the $\preceq_k$-least model of $\mathcal{P}$, and the *Well-Founded model* ($WF_{\mathcal{P}}$) [10,26]. It is well-know that the $WF_{\mathcal{P}}$ is more informative (provides more knowledge) than $KK_{\mathcal{P}}$. For the definition of the well-founded semantics over bilattices refer to [10,17]. It is the generalization of the classical well-founded semantics to bilattices. We obtain it as a special case of the AWA, too. Furthermore, we note that the existence and uniqueness of $KK_{\mathcal{P}}$ is guaranteed by the fixed-point characterization based on the $\preceq_k$-monotone function $\Phi_{\mathcal{P}}$: for an interpretation I, for any ground atom A with (unique) $A \leftarrow \varphi \in \mathcal{P}^*$, $\Phi_{\mathcal{P}}(I)(A) = I(\varphi)$. Then all models of $\mathcal{P}$ are fixed-points of $\Phi_{\mathcal{P}}$ and vice-versa, and $KK_{\mathcal{P}}$ can be computed in the usual way by iterating $\Phi_{\mathcal{P}}$ over $I_\bot$.

Classical Logic Programs. In classical logic programs the body is a conjunction of literals, i.e., for $A \leftarrow \varphi \in \mathcal{P}^*$ (except for the case $A \leftarrow \mathtt{f} \in \mathcal{P}^*$) $\varphi = \varphi_1 \vee \ldots \vee \varphi_n$ and $\varphi_i = L_{i_1} \wedge \ldots \wedge L_{i_n}$. For a set of literals X, with $\neg.X$ we indicate the set $\{\neg L \colon L \in X\}$, where for any atom A, $\neg\neg A$ is replaced with A. A classical interpretation (total or partial) can be represented as a consistent set of literals, i.e., $I \subseteq B_{\mathcal{P}} \cup \neg.B_{\mathcal{P}}$ and for all atoms A, $\{A, \neg A\} \not\subseteq I$. Of course, the opposite is also true, i.e., a consistent set of literals can straightforwardly be turned into an interpretation over $\mathcal{FOUR}$.

The classical WF semantics has been defined in terms of the well-known notion of *unfounded set* (see e.g., [14,26]), which identifies the set of atoms that can safely be assumed false if the current information about $\mathcal{P}$ is given by an interpretation I. Indeed, given a partial classical interpretation I and a classical logic program $\mathcal{P}$, a set of ground atoms $X \subseteq B_{\mathcal{P}}$ is an *unfounded set* (i.e.,, the atoms in X can be assumed as false) for $\mathcal{P}$ w.r.t. I iff for each atom $A \in X$, if $A \leftarrow \varphi \in \mathcal{P}^*$, where $\varphi = \varphi_1 \vee \ldots \vee \varphi_n$ and $\varphi_i = L_{i_1} \wedge \ldots \wedge L_{i_n}$, then φ_i is false either w.r.t. I or w.r.t. $\neg.X$, for all $1 \leq i \leq n$. The *greatest unfounded set* for $\mathcal{P}$ w.r.t. I (which exists) is denoted by $U_{\mathcal{P}}(I)$. Then, the well-founded semantics $WF_{\mathcal{P}}$ is defined to be [14]: $WF_{\mathcal{P}}$ = " $\preceq_k$-least model I of $\mathcal{P}$ such that $\neg.U_{\mathcal{P}}(I) \subseteq I$". As we will see next, the AWA generalizes this notion.

3 The AWA in Logic Programming

A *hypothesis* (denoted H) is always an interpretation over a bilattice and represents our default assumption over the world. The principle underlying the *Any-World Assumption* (AWA) is to regard an hypothesis H as an additional source of default information to be used to complete the implicit knowledge provided by a logic program. The AWA H dictates that any atom A, whose truth-value cannot be inferred from the facts and rules, is assigned to the default truth value $H(A)$. For comparison, under the CWA, $H = I_{\mathtt{f}}$ is assumed, while under the OWA, $H = I_\bot$ is assumed. Also note that any ground atom A not appearing in the head of any rule and, thus, not derivable, is mapped (up to now) into 'false'. Now, according to the AWA, any such atom A should be mapped

into $H(A)$. *If not specified otherwise, we change Point 3. of the definition of* $\mathcal{P}^*$ *by adding* $A \leftarrow H(A)$ *to* $\mathcal{P}^*$. It should be noted that this implicitly affects also all definitions based on $\mathcal{P}^*$, e.g., the definitions of model and that of $\Phi_\mathcal{P}$ (which now maps such atoms into $H(A)$ rather than into $\mathtt{f}$). To emphasize the impact of H to $\Phi_\mathcal{P}$, we denote the immediate consequence operator with $\Phi^H_\mathcal{P}$ in place of $\Phi_\mathcal{P}$. Now, we proceed in two steps.

The support. At first, we introduce the notion of *support*, denoted $s^H_\mathcal{P}(I)$. The support is a generalization of the notion of unfounded sets. Indeed, $s^H_\mathcal{P}(I)$ determines the amount of default information, taken from H, that can safely be joined to I. The support generalizes the notion of unfounded sets as it turns out that for classical logic programs $\mathcal{P}$ and $H = I_\mathtt{f}$ (see Table 1), $s^H_\mathcal{P}(I) = \neg.U_\mathcal{P}(I)$ [16]. The principle underlying the support can be explained as follows. Consider a ground atom A and the rule $A \leftarrow \varphi \in \mathcal{P}^*$, an interpretation I, which is our current knowledge about $\mathcal{P}$, and a hypothesis H. We would like to determine how much default knowledge can be 'safely' taken from H to complete I. So, let us assume that $J \preceq_k H$ amounts to the default knowledge taken from H. $J(A)$ is the default information provided by J to the atom A. The completion of I with J is the interpretation $I \oplus J$. In order to accept this completion, we have to ensure that at least the assumed knowledge $J(A)$ is entailed by $\mathcal{P}$ w.r.t. the completed interpretation $I \oplus J$, i.e., for $A \leftarrow \varphi \in \mathcal{P}^*$, $J(A) \preceq_k (I \oplus J)(\varphi) = \Phi^H_\mathcal{P}(I \oplus J)(A)$ should hold. Therefore, we say that an interpretation J is *safe* w.r.t. $\mathcal{P}$, I and H iff $\preceq_k H$ and $J \preceq_k \Phi^H_\mathcal{P}(I \oplus J)$. Note that safe interpretations correspond to unfounded sets for classical logic programs [16]. Furthermore, like for unfounded sets, among all possible safe interpretations, we are interested in the $\preceq_k$-maximal (which exists and is unique). The $\preceq_k$-greatest safe interpretation is called the *support* provided by H to $\mathcal{P}$ w.r.t. I and is denoted by $s^H_\mathcal{P}(I)$. Table 1 reports the support for the logic program of Example 1. Note that by definition under the OWA $H = I_\bot$, $s^H_\mathcal{P}(I) = I_\bot$ holds, as expected, while for classical logic programs $s^H_\mathcal{P}(I) = \neg.U_\mathcal{P}(I)$, for $H = I_\mathtt{f}$. In summary, the support is an extension of the notion of unfounded sets (i) to logic programming over bilattices; and to (ii) arbitrary default assumptions H. Finally, we also recall that the support can effectively be computed as the iterated fixed-point of the $\preceq_k$-monotone function $\sigma^{I,H}_\mathcal{P}(J) = H \otimes \Phi^H_\mathcal{P}(I \oplus J)$. Indeed, [16] shows that the iterated sequence of interpretations J_i below is $\preceq_k$-monotone decreasing and reaches a fixed-point, $J_\lambda = s^H_\mathcal{P}(I)$, for a limit ordinal λ, where $J_0 = H$, $J_{i+1} = \sigma^{I,H}_\mathcal{P}(J_i)$, $J_\lambda = \inf_{i<\lambda} \sigma^{I,H}_\mathcal{P}(J_i)$.

H-models. At second, among all models of a program $\mathcal{P}$, let us consider those models, which $\preceq_k$-subsume their own support. That is, we say that an interpretation I is a *H-model* of $\mathcal{P}$ iff $I \models \mathcal{P}$ and $s^H_\mathcal{P}(I) \preceq_k I$. The $\preceq_k$-least H-model is called *H-founded model*, and is denoted with $HF_\mathcal{P}$. H-models have interesting properties [16].

Proposition 1 ([16]). *I is a H-model of $\mathcal{P}$ iff $I = \Phi^H_\mathcal{P}(I \oplus s^H_\mathcal{P}(I))$.*

From a fixed-point characterization point of view, it follows immediately that the set of H-models can be identified by the fixed-points of the $\preceq_k$-monotone immediate consequence operator:

$$\Pi^H_\mathcal{P}(I) = \Phi^H_\mathcal{P}(I \oplus s^H_\mathcal{P}(I)) \,.$$

This guarantees the existence and uniqueness of the $\preceq_k$-least fixed-point of $\Pi^H_{\mathcal{P}}(I)$, i.e., the H-founded model of a program $\mathcal{P}$.

Note that the definition of H-founded model is nothing else than a generalization from the classical setting to bilattices of the notion of well-founded model (recall that the well-founded model is the least model satisfying $\neg.U_{\mathcal{P}}(I) \subseteq I$ [14], which is a special case of the definition of H-founded model). We conclude by remarking that [16] also generalizes the stable model semantics to the AWA.

Example 2. Consider Example 1 and Table 1. Given the hypothesis H_i described in the tables (note that $H_1 = H_2 = I_{\mathtt{f}}$, i.e., the CWA is assumed), we observe that $s^{H_i}_{\mathcal{P}}(I_i) \preceq_k I_i$ for $i = 2, 3$ and, thus, both I_2 and I_3 are H-models, while I_1 is not. Furthermore, it can be verified that both I_2 and I_3 are also H-founded models and that I_2 corresponds to the classical well-founded semantics, as expected.

We refer the reader to [16,24] for some applications of the AWA. For the sake of illustrative purposes, we recall the following example: a rule expressing the fact that *a car may cross railway tracks if there is no crossing train* may be represented by $\mathtt{Cross\_railway} \leftarrow \neg\mathtt{Train\_is\_comming}$. In this situation, in order to safely cross the railway there should be explicit evidence that the train is not coming and, thus, we assume by default that $H(\mathtt{Train\_is\_comming}) = \bot$ (i.e., the atom is interpreted according to the OWA) and $H(\mathtt{Cross\_railway}) = \mathtt{f}$ (i.e., the CWA is assumed), for safety.

Another example is the case where we also want to express *default statements* of the form *normally, unless something abnormal holds, then φ implies A*. Such statements were the main motivation for non-monotonic logics like *Default Logic* [22], Autoepistemic Logic [8,18,20,21] and *Circumscription* [19]. We can formulate such a statement in a natural way, using *abnormality theories*, as $A \leftarrow \varphi \wedge \neg Ab$ and $Ab \leftarrow \neg A$, where Ab stands for *abnormality*, and then consider the hypothesis $H(Ab) = \mathtt{f}$, i.e., by default there are no abnormal objects.

4 Top-Down Query Answering

A *query* is an atom Q (*query atom*) of the form $q(\mathbf{x})$, intended as a question about the truth degree of all the instances of Q in the intended model of $\mathcal{P}$. We also allow a query to be a *set* $\{Q_1, \ldots, Q_n\}$ of query atoms. In that latter case we ask about the truth degree of all instances of the atoms Q_i in the intended model.

The procedure we devise in this paper is a generalization of the procedure presented in [24]. We anticipate that the main reason why the procedure in [24] is not suitable to be used for computing all answers to a query Q, given $\mathcal{P}$, is that (i) [24] relies on $\mathcal{P}$'s grounded version $\mathcal{P}^*$, which may be rather huge (exponential with respect to $|\mathcal{P}|$, in general) in applications with many facts; (ii) [24] answers ground queries only. Strictly speaking, [24] can compute all answers of a query atom $q(\mathbf{x})$ by submitting as query the set of all ground instances $q(\mathbf{c})$. This is clearly not feasible if the Herbrand universe is large. The procedure presented here does not require grounding.

In the following, we assume that a logic program $\mathcal{P}$ is made out of an *extensional database* (EDB), $\mathcal{P}_E$, and an *intensional database* (IDB), $\mathcal{P}_I$. The extensional database

is a set of facts of the form $r(c_1, \ldots, c_n) \leftarrow b$, where $r(c_1, \ldots, c_n)$ is a ground atom and b is a truth value. For convenience, for each n-ary extensional predicate r, we represent the facts $r(c_1, \ldots, c_n) \leftarrow b$ in $\mathcal{P}$ by means of a relational $n+1$-ary table tab_r, containing the records $\langle c_1, \ldots, c_n, b\rangle$. Thus, the table contains all the instances of r together with their degrees. We assume that there cannot be two records $\langle c_1, \ldots, c_n, b_1\rangle$ and $\langle c_1, \ldots, c_n, b_2\rangle$ in tab_r with $b_1 \neq b_2$. The intensional database is a set of rules for the form $p(\mathbf{x}) \leftarrow \varphi(\mathbf{x}, \mathbf{y})$ in which the predicates occurring in the extensional database (called *extensional predicates*) do not occur in the head of rules of the intensional database. Essentially, we do not allow that the fact predicates occurring in $\mathcal{P}_E$ can be redefined by $\mathcal{P}_I$. We also assume that the *intensional predicate* symbol p occurs in the head of at most one rule in the intensional database. Due to the expressiveness of rule bodies, it is not difficult to see that logic programs can be put into this form.

For an atom A of the form $p(\mathbf{x})$, an *answer* for p is a pair $\langle \theta, b\rangle$, where $\theta = \{\mathbf{x}/\mathbf{c}\}$ is a substitution of the variables $\mathbf{x}$ in $p(\mathbf{x})$ with the constants in $\mathbf{c}$ and $b \in L$ is a truth degree. We say that the answer $\langle \theta, b\rangle$ is *correct* for p with respect to the intended model I of $\mathcal{P}$ iff $I(p(\mathbf{c})) = b$. That is, by substituting the variables in $\mathbf{x}$ using θ, the evaluation of the query in the intended model is b. An *answer set* for p is a set of answers for p. Of course, our goal is to determine the set of all correct answers for the query Q. For a given n-ary predicate p and a set of answers Δ_p of p, for convenience we represent Δ_p as an $n+1$-ary table tab_{Δ_p}, containing the records $\langle c_1, \ldots, c_n, b\rangle$.

Given two answers $\delta_1 = \langle \theta, b_1\rangle$ and $\delta_2 = \langle \theta, b_2\rangle$ for the same atom P, we define $\delta_1 \preceq_k \delta_2$ ($\delta_1 \succeq_k \delta_2$) iff $b_1 \preceq_k b_2$ ($b_1 \succeq_k b_2$). We write $\delta_1 \prec_k \delta_2$ ($\delta_1 \succ_k \delta_2$) iff $b_1 \prec_k b_2$ ($b_1 \succ_k b_2$). If Δ_p^1 and Δ_p^2 are two sets of answers for p, we write $\Delta_p^1 \preceq_k \Delta_p^2$ ($\Delta_p^1 \succeq_k \Delta_p^2$) iff for all $\delta_1 \in \Delta_p^1$ there is $\delta_2 \in \Delta_p^2$ such that $\delta_1 \preceq_k \delta_2$ ($\delta_1 \succeq_k \delta_2$). We write $\Delta_p^1 \prec_k \Delta_p^2$ ($\Delta_p^1 \succ_k \Delta_p^2$) iff $\Delta_p^1 \preceq_k \Delta_p^2$ ($\Delta_p^1 \succeq_k \Delta_p^2$) and there is $\delta_2 \in \Delta_p^2$ such that for no $\delta_1 \in \Delta_p^1$, $\delta_2 \preceq_k \delta_1$ ($\delta_2 \succeq_k \delta_1$) holds.

We present now our top-down procedure tailored to compute all correct answer of a query Q in the intended model. The basic idea of our procedure is to try to collect, during the computation, all correct answers incrementally together. At first, consider a general rule of the form $p(\mathbf{x}) \leftarrow \varphi(\mathbf{x}, \mathbf{y})$. We note that $\varphi(\mathbf{x}, \mathbf{y})$ depends on a computable function f and the predicates $p_1, \ldots, p_k$, which occur in the rule body $\varphi(\mathbf{x}, \mathbf{y})$. Assume that $\Delta_{p_1}, \ldots, \Delta_{p_k}$ are the answers collected so far for the predicates $p_1, \ldots, p_k$. Let us consider a procedure $eval(p, \Delta_{p_1}, \ldots, \Delta_{p_k})$, which computes the set of answers $\langle \{\mathbf{x}/\mathbf{c}\}, b\rangle$ of p, by evaluating the body $\varphi(\mathbf{x}, \mathbf{y})$ over the data provided by $\Delta_{p_1}, \ldots, \Delta_{p_k}$. Formally, let H be a hypothesis, let I^H be an interpretation restricted to the predicates $p_1, \ldots, p_k$ and tuples such that for all n_i-ary predicates p_i,

$$I^H(p_i(\mathbf{c})) = \begin{cases} b, & \text{if } \langle \mathbf{c}, b\rangle \in tab_{\Delta_{p_i}} \\ H(p_i(\mathbf{c})) & \text{if } p_i \text{ is an extensional predicate and } \langle \mathbf{c}, b\rangle \notin tab_{\Delta_{p_i}} \\ \bot & \text{otherwise .} \end{cases}$$

The intuition in the definition above is that to an atom $p_i(\mathbf{c})$ we assign the current truth value if this truth value is known. Otherwise, we assign to it the default truth value taken from the hypothesis (if p_i is an extensional predicate). Then

$$eval(p, H, \Delta_{p_1}, \ldots, \Delta_{p_k}) = \{\langle \{\mathbf{x}/\mathbf{c}\}, b\rangle \mid b = \bigvee_{\mathbf{c}'} I^H(\varphi(\mathbf{c}, \mathbf{c}')), b \neq \bot\} ,$$

where $\mathbf{c}'$ is a tuple of constants occurring in $\bigcup_i \Delta_{p_i}$. We omit to report the tuple whose degree is $\bot$. The disjunction $\bigvee_{\mathbf{c}'}$ is required as the free variables $\mathbf{y}$ in $\varphi(\mathbf{x}, \mathbf{y})$ may be seen as existentially quantified.

Example 3. Consider $\mathcal{P} = \{p(x) \leftarrow q(x,y), q(a,b) \leftarrow \mathtt{f}, q(a,c) \leftarrow \mathtt{t}\}$. Assume $\Delta_q = \{\langle (a,b), \mathtt{f}\rangle, \langle (a,c), \mathtt{t}\rangle\}$. Then $eval(p, \Delta_q) = \{\langle a, \mathtt{t}\rangle\}$, which amounts to evaluate $q(a,b) \vee q(a,c)$.

We are not going to further investigate the implementation details of the $eval(p, H, \Delta_{p_1}, \ldots, \Delta_{p_k})$ procedure, though it has to be carefully written to minimize the number of table look-ups and relational algebraic operations such as joins. It can be obtained by means of a combination of SQL statements over the tables and the application of the truth combination functions occurring in the rule body of p. We point out that $eval(p, H, \Delta_{p_1}, \ldots, \Delta_{p_k})$ can also be seen as a query to a database made out by the relations $tab_{\Delta_{p_1}}, \ldots, tab_{\Delta_{p_k}}$ and that any successive evaluation step corresponds to the execution of the *same* query over an updated database. We refer the reader to e.g., [6,7] concerning the problem of repeatedly evaluating the same query to a database that is being updated between successive query requests. In this situation, it may be possible to use the difference between successive database states and the answer to the query in one state to reduce the cost of evaluating the query in the next state.

Query answering: Kripke-Kleene semantics. We start showing how to compute all answers with respect to the Kripke-Kleene semantics, i.e., the $\preceq_k$-least fixed-point of $\Phi^H_{\mathcal{P}}$. The procedure is detailed in Table 2. Assume, we are interested in determining all correct answers of $q(\mathbf{x})$ w.r.t. the Kripke-Kleene semantics. We call the procedure with $Answer(\mathcal{P}, Q, H)$. We start with putting the predicate symbols $q \in Q$ in the *active* list of predicate symbols $\mathtt{A}$. At each iteration step (step 2) we select a new predicate p from the queue $\mathtt{A}$ and evaluate it using the $eval$ function with respect to the answers gathered so far (steps 4 or 5). If the evaluation leads to a better answer set for p (step 6), we update the current answer set $\mathtt{v}(p)$ and add all predicates p', whose rule body contains p (the parents of p), to the queue $\mathtt{A}$, i.e., all predicate symbols that might depend on p are put in the active set to be examined. At some point (even if cyclic definitions are present) the active list will become empty and we have actually found all correct answers of $q(\mathbf{x})$. The procedure in Table 2 uses some auxiliary functions and data structures: (i) for predicate symbol p_i, $\mathtt{s}(p_i)$ is the set of predicate symbols occurring in the rule body

Table 2. General top-down algorithm

Procedure $Answer(\mathcal{P}, Q, H)$
Input: Logic program $\mathcal{P}$, set Q of query predicate symbols, hypothesis H;
Output: Mapping $\mathtt{v}$ containing all correct answers of predicates in Q w.r.t. lfp($\Phi^H_{\mathcal{P}}$)
1. $\mathtt{A} := Q$, $\mathtt{dg} := Q$, $\mathtt{in} := \emptyset$, **for all** predicate symbols p in $\mathcal{P}$ **do** $\mathtt{v}(p) = \emptyset$, $\mathtt{exp}(p) = \mathtt{false}$
2. **while** $\mathtt{A} \neq \emptyset$ **do**
3. **select** $p_i \in \mathtt{A}$, $\mathtt{A} := \mathtt{A} \setminus \{p_i\}$, $\mathtt{dg} := \mathtt{dg} \cup \mathtt{s}(p_i)$
4. **if** (p_i extensional predicate) $\wedge$ ($\mathtt{v}(p_i) = \emptyset$) **then** $\mathtt{v}(p_i) := tab_{p_i}$
5. **if** (p_i intensional predicate) **then** $\Delta_{p_i} := eval(p_i, H, \mathtt{v}(p_{i_1}), \ldots, \mathtt{v}(p_{i_{k_i}}))$
6. **if** $\mathtt{v}(p_i) \prec_k \Delta_{p_i}$ **then** $\mathtt{v}(p_i) := \Delta_{p_i}$, $\mathtt{A} := \mathtt{A} \cup (\mathtt{p}(p_i) \cap \mathtt{dg})$
7. **if not** $\mathtt{exp}(p_i)$ **then** $\mathtt{exp}(p_i) = \mathtt{true}$, $\mathtt{A} := \mathtt{A} \cup (\mathtt{s}(p_i) \setminus \mathtt{in})$, $\mathtt{in} := \mathtt{in} \cup \mathtt{s}(p_i)$
endwhile

of p_i, i.e., the *sons* of p_i; (ii) for predicate symbol p_i, $\mathtt{p}(p_i) = \{p_j : p_i \in \mathtt{s}(p_j)\}$, i.e., the *parents* of p_i; (iii) in step 5, $p_{i_1}, \ldots, p_{i_{k_i}}$ are all predicate symbols occurring in the rule body of p_i, i.e., the sons $\mathtt{s}(p_i) = \{p_{i_1}, \ldots, p_{i_{k_i}}\}$ of p_i; (iv) the variable `dg` collects the predicate symbols that may influence the result of the query predicates; (v) the array variable `exp` traces the rule bodies that have been "expanded" (the predicate symbols occurring in the rule body are put into the active list); (vi) the variable `in` keeps track of the predicate symbols that have been put into the active list so far due to an expansion (to avoid, to put the same predicate symbol multiple times in the active list due to rule body expansion).

Example 4. Consider Example 1. Let us consider the hypothesis $H = I_\bot$ (i.e., the OWA). The extensional database is shown in the relational table $tab_r = \{\langle a, \mathtt{t}\rangle, \langle b, \mathtt{f}\rangle\}$. Of course, tab_r is also the set tab_{Δ_r} of correct answers of predicate r, while it can be verified (by a straightforward bottom-up fixed-point computation iterating $\Phi^H_\mathcal{P}$ over $I_\bot$) that the set of correct answers of predicate q is given by: $\Delta_q = \{\langle b, \mathtt{t}\rangle\}$. We do not report the tuple $\langle a, \bot\rangle$, as if $\mathbf{c}$ does not occur in an answer set Δ then its truth degree is assumed to be $\bot$. We next show the computation of $Answer(\mathcal{P}, \{q\}, H)$. The execution is shown below reporting also Δ_{p_i} and $\mathtt{v}(p_i)$ at each iteration i. Each line is a sequence of steps in the 'while loop'. What is left unchanged is not reported.

1. $\mathtt{A} := \{q\}, p_i := q, \mathtt{A} := \emptyset, \mathtt{dg} := \{q, r\}, \Delta_q := \emptyset$ $\mathtt{exp}(q) := 1, \mathtt{A} := \{q, r\}, \mathtt{in} := \{q, r\}$	$Iter\ i$	Δ_{p_i}	$\mathtt{v}(p_i)$
2. $p_i := q, \mathtt{A} := \{r\}, \Delta_q := \emptyset$	0.	–	$\mathtt{v}(p_i) = \emptyset$
3. $p_i := r, \mathtt{A} := \emptyset, \mathtt{v}(r) \prec_k \Delta_r, \mathtt{v}(r) := \Delta_r, \mathtt{A} := \{q\}, \mathtt{exp}(r) := 1$	1.	$\Delta_q = \emptyset$	–
4. $p_i := q, \mathtt{A} := \emptyset, \mathtt{v}(q) \prec_k \Delta_q, \mathtt{v}(q) := \Delta_q, \mathtt{A} := \{q\}$	2.	$\Delta_q = \emptyset$	–
5. $p_i := q, \mathtt{A} := \emptyset, \Delta_q = \mathtt{v}(q)$	3.	$\Delta_r = \{\langle a, \mathtt{t}\rangle, \langle b, \mathtt{f}\rangle\}$	$\mathtt{v}(r) = \Delta_r$
6. `stop. return` $\mathtt{v}(q)$	4.	$\Delta_q = \{\langle b, \mathtt{t}\rangle\}$	$\mathtt{v}(q) = \Delta_q$
	5.	$\Delta_q = \{\langle b, \mathtt{t}\rangle\}$	–

It can be shown that the procedure $Answer$ behaves as expected.

Proposition 2. *There is a limit ordinal λ such that after $|\lambda|$ steps $Answer(\mathcal{P}, Q, H)$ returns the set of all correct answers of $\mathcal{P}$ with respect to the predicates in Q and the Kripke-Kleene semantics under hypothesis H.*

Computatonally, it is well known that, despite the Herbrand base is finite, many-valued logic programs may not have a finite bottom-up least model computation and our framework does inherit the same problems as well. There are, however several useful options to guarantee termination of the top-down procedure such as: (i) the bilattice has a finite number of truth values; (ii) the truth combination function f in a rule body is a upper-bounded, i.e., $f(x_1, \ldots, x_n) \preceq_t x_i$, for all i (e.g., a t-norm satisfies this condition); (iii) if B is $[0, 1] \times [0, 1]$, all f are continuos then for any $\epsilon > 0$ the procedure stops after a finite number of steps such that the final truth of an atoms diverges from the actual value at most ϵ (this result is similar to the one established in [27]).

Query answering: H-founded semantics. As we have seen, the H-founded model of a logic program $\mathcal{P}$ is the $\preceq_k$-least fixed-point of the operator $\Pi^H_\mathcal{P}$ (see Proposition 1) and the support $s^H_\mathcal{P}(I)$ coincides with the iterated fixed-point of the function $\sigma^{I,H}_\mathcal{P}(J)$ beginning the computation with H. In the following, we show how we can slightly change the $Answer$ procedure to compute the support. That is, we want a top-down procedure that, for a set of atoms $p(\mathbf{x})$, computes all answers $\langle\{\mathbf{x}/\mathbf{c}\}, b\rangle$ such that $s^H_\mathcal{P}(I)(p(\mathbf{c})) = b$.

So, let $Support(\mathcal{P}, Q, H, I)$ be the procedure, which is as the $Answer$ procedure except that:

- Step 1 is replaced with

  $\mathcal{P} := \mathcal{P}_I^H$, $\mathtt{A} := Q$, $\mathtt{dg} := Q$, $\mathtt{in} := \emptyset$, **for all** predicate symbols p in $\mathcal{P}$ **do** $\mathtt{v}(p) = \emptyset$, $\mathtt{exp}(p) = \mathtt{false}$

  where logic program $\mathcal{P}_I^H$ is obtained from $\mathcal{P}$ in the following way:
  - for each intensional predicate p in $\mathcal{P}$, replace the rule $p(\mathbf{x}) \leftarrow \varphi(\mathbf{x}, \mathbf{y})$ in $\mathcal{P}$ with the rule
  $$p(\mathbf{x}) \leftarrow H(p)(\mathbf{x}) \otimes (I(p_\varphi)(\mathbf{x}) \oplus \varphi(\mathbf{x}, \mathbf{y})) \,. \tag{1}$$
  With $H(p)(\mathbf{x})$ we mean a built-in predicate that given a substitution $\mathbf{c}$ for $\mathbf{x}$, returns $H(p(\mathbf{c}))$. This can easily be encoded in the semantics, which we omit. The case $I(p_\varphi)(\mathbf{x})$ is similar: $I(p_\varphi)(\mathbf{x})$ is a built-in predicate that given a substitution $\mathbf{c}$ for $\mathbf{x}$, returns $\bigvee_{\mathbf{c}'} I(\varphi(\mathbf{c}, \mathbf{c}'))$.
  - for each extensional predicate r in $\mathcal{P}$, replace the rule $r(\mathbf{c}) \leftarrow b$ in $\mathcal{P}$ with the rule
  $$r(\mathbf{c}) \leftarrow b' \,, \tag{2}$$
  where b' is the truth value $b' = H(r(\mathbf{c})) \otimes b$.

  We point out that the rules above are the result of applying $\sigma_{\mathcal{P}}^{I,H}$ to the support $s_{\mathcal{P}}^H(I)$ and to all rules:
  $$\begin{aligned} s_{\mathcal{P}}^H(I)(p(\mathbf{c})) &= [H \otimes \Phi_{\mathcal{P}}^H(I \oplus s_{\mathcal{P}}^H(I))](p(\mathbf{c})) = H(p(\mathbf{c})) \otimes [I \oplus s_{\mathcal{P}}^H(I)](\textstyle\bigvee_{\mathbf{c}'} \varphi(\mathbf{c}, \mathbf{c}')) \\ &= H(p(\mathbf{c})) \otimes (I(\textstyle\bigvee_{\mathbf{c}'} \varphi(\mathbf{c}, \mathbf{c}')) \oplus s_{\mathcal{P}}^H(I)(\textstyle\bigvee_{\mathbf{c}'} \varphi(\mathbf{c}, \mathbf{c}'))) \\ &= H(p(\mathbf{c})) \otimes (\textstyle\bigvee_{\mathbf{c}'} I(\varphi(\mathbf{c}, \mathbf{c}')) \oplus \bigvee_{\mathbf{c}'} s_{\mathcal{P}}^H(I)(\varphi(\mathbf{c}, \mathbf{c}'))) \,. \end{aligned}$$
  Since the above equation holds for all predicates p and all $\mathbf{c}$, we get rules (1) and (2). Built-in predicates do not count as sons and, thus, do not appear in the $\mathtt{A}$, $\mathtt{s}$, $\mathtt{p}$, $\mathtt{v}$, $\mathtt{in}$, $\mathtt{dg}$ variables.

- Step 6 is replaced with

  if $\mathtt{v}(p_i) \succ_k \Delta_{p_i}$ **then** $\mathtt{v}(p_i) := \Delta_{p_i}$, $\mathtt{A} := \mathtt{A} \cup (\mathtt{p}(p_i) \cap \mathtt{dg})$ **fi**

  Essentially, in Step 6 we replace $\prec_k$ with $\succ_k$. This modification is motivated by the fact that during the computation of the support, Δ_{p_i} is now decreasing in the knowledge order $\preceq_k$.

Example 5. Consider Example 1, interpretation I_2 and hypothesis H_2. We have seen that I_2 is the H-founded model of $\mathcal{P}$ w.r.t. H_2 and corresponds to the well-founded semantics of $\mathcal{P}$. We next want to show the computation of $Support(\mathcal{P}, \{q, r\}, H_2, I_2)$. We first determine $\mathcal{P}_{I_2}^{H_2}$. As predicate p does not play any role in the computation, we report the modified rule for predicate q and r only. $\mathcal{P}_{I_2}^{H_2}$ related to q and r is $\{q(x) \leftarrow H_2(q)(x) \otimes (I_2(q_\varphi)(x) \oplus (q(x) \vee \neg r(x))), r(a) \leftarrow \bot, r(b) \leftarrow \mathtt{f}\} \subseteq \mathcal{P}_{I_2}^{H_2}$.

We recall that $H_2(q)(a) = H_2(q)(b) = \mathtt{f}$ and that $I_2(q_\varphi)(a) = I_2(q(a) \vee \neg r(a)) = \mathtt{f}$, while $I_2(q_\varphi)(b) = \mathtt{t}$. Then, it can be verified that (by a straightforward fixed-point computation iterating $\sigma_{\mathcal{P}}^{I,H}$ starting with H_2) that the set of correct answers of predicate q, r of $\mathcal{P}$ w.r.t. $s_{\mathcal{P}}^{H_2}(I_2)$ are: $\Delta_q = \{\langle a, \mathtt{f}\rangle\}, \Delta_r = \{\langle b, \mathtt{f}\rangle\}$.

Below is a sequence of $Support(\mathcal{P}, \{q, r\}, H_2, I_2)$, returning the expected values.

1. $\mathtt{A} := \{q, r\}, p_i := q, \mathtt{A} := \{r\}, \mathtt{dg} := \{q, r\}, \Delta_q \succ_k \mathtt{v}(q),$ $\mathtt{exp}(q) := 1, \mathtt{A} := \{r, q\}, \mathtt{in} := \{q, r\}$ 2. $p_i := r, \mathtt{A} := \{q\}, \mathtt{v}(r) \succ_k \Delta_r, \mathtt{v}(r) := \Delta_r, \mathtt{exp}(r) := 1$ 3. $p_i := q, \mathtt{A} := \emptyset, \Delta_q = \mathtt{v}(q)$ 4. $\mathtt{stop}.\ \mathtt{return}\ \mathtt{v}(q)$	$Iter\ i$	Δ_{p_i}	$\mathtt{v}(p_i)$
	0.	–	$\mathtt{v}(p_i) = \emptyset$
	1.	$\Delta_q = \{\langle a, \mathtt{f}\rangle\}$	$\mathtt{v}(q) = \Delta_q$
	2.	$\Delta_r = \{\langle b, \mathtt{f}\rangle\}$	$\mathtt{v}(r) = \Delta_r$
	3.	$\Delta_q = \{\langle a, \mathtt{f}\rangle\}$	–

It can then be shown that:

Proposition 3. *There is a limit ordinal λ such that after $|\lambda|$ steps $Support(\mathcal{P}, Q, H, I)$ returns the set of all correct answers of $\mathcal{P}$ with respect to the predicates in Q and the support $s^H_{\mathcal{P}}(I)$.*

We are now ready to define the top-down procedure $Answer_{HF}(\mathcal{P}, Q, H)$, which computes all correct answers to a query Q under the H-founded semantics. We define $Answer_{HF}(\mathcal{P}, Q, H)$ as $Answer(\mathcal{P}, Q, H)$, except that Step 5 is replaced with the statements

5. **if** (p_i intensional predicate) **then**
5.1. $\mathtt{Q}' := \mathtt{s}(p_i)$;
5.2. $\mathtt{I} := \mathtt{v}$;
5.3. $\mathtt{supp} := Support(\mathcal{P}, \mathtt{Q}', H, \mathtt{I})$;
5.4. $\mathtt{v}' := \mathtt{I} \oplus \mathtt{supp}$;
5.5. $\Delta_{p_i} := eval(p_i, H, \mathtt{v}'(p_{i_1}), ..., \mathtt{v}'(p_{i_{k_i}}))$ **fi**

These steps correspond to the application of the $\Pi^H_{\mathcal{P}}(I) = \Phi^H_{\mathcal{P}}(I \oplus s^H_{\mathcal{P}}(I))$ operator to p_i. Indeed, at first we ask about all the correct answers of the predicates occurring in the body of p_i w.r.t. the support and the current interpretation $\mathtt{I} := \mathtt{v}$ (Steps 5.1 - 5.3). The variable `supp` holds these answers. Then we join them with `I`, i.e., we compute $I \oplus s^H_{\mathcal{P}}(I)$ (Step 5.4), where this latter is defined pointwise: (i) $\mathtt{v}' = \mathtt{v}_1 \oplus \mathtt{v}_2$ iff for all p, $\mathtt{v}'(p) = \mathtt{v}_1(p) \oplus \mathtt{v}_2(p) = \{\langle\theta, b\rangle \mid \langle\theta, b_1\rangle \in \mathtt{v}_1(p), \langle\theta, b_2\rangle \in \mathtt{v}_2(p), b = b_1 \oplus b_2\}$ (if $\langle\theta, b_i\rangle \notin \mathtt{v}_i(p)$ then $b_i = \bot$ is assumed). Finally, we evaluate the body of p_i with respect to $I \oplus s^H_{\mathcal{P}}(I)$ (Step 5.5), i.e., apply $\Phi^H_{\mathcal{P}}(I \oplus s^H_{\mathcal{P}}(I))$.

Example 6. Consider Example 1 and hypothesis H_2 (i.e., the CWA). Let us compute all correct answers to the query $q(x)$ w.r.t. the well-founded semantics. As the interpretation I_2 in Example 6 is the well-founded model (i.e., H_2-founded model), we expect to retrieve $\Delta_q = \{\langle a, \mathtt{f}\rangle, \langle b, \mathtt{t}\rangle\}$. Below is the computation of $Answer_{HF}(\mathcal{P}, \{q\}, H_2)$.

1. $\mathtt{A} := \{q\}, p_i := q, \mathtt{A} := \emptyset, \mathtt{dg} := \{q, r\}, \mathtt{supp} := \{\langle r(b), \mathtt{f}\rangle\}$,
   $\mathtt{v}' := \{\langle r(b), \mathtt{f}\rangle\}$,
   $\mathtt{v}(q) \prec_k \Delta_q, \mathtt{exp}(q) := 1, \mathtt{A} := \{q, r\}, \mathtt{in} := \{q, r\}$
2. $p_i := q, \mathtt{A} := \{r\}, \mathtt{supp} := \{\langle r(b), \mathtt{f}\rangle\}$,
   $\mathtt{v}' := \{\langle q(b), \mathtt{t}\rangle, \langle r(b), \mathtt{f}\rangle\}, \Delta_q = \mathtt{v}(q)$
3. $p_i := r, \mathtt{A} := \emptyset, \mathtt{supp} := \{\langle r(b), \mathtt{f}\rangle\}, \mathtt{v}' := \{\langle q(b), \mathtt{t}\rangle, \langle r(b), \mathtt{f}\rangle\}$,
   $\mathtt{v}(r) \prec_k \Delta_r, \mathtt{v}(r) := \Delta_r, \mathtt{A} := \{q\}, \mathtt{exp}(r) := 1$
4. $p_i := q, \mathtt{A} := \emptyset, \mathtt{supp} := \{\langle q(a), \mathtt{f}\rangle, \langle r(b), \mathtt{f}\rangle\}$,
   $\mathtt{v}' := \{\langle q(a), \mathtt{f}\rangle, \langle q(b), \mathtt{t}\rangle, \langle r(a), \mathtt{t}\rangle, \langle r(b), \mathtt{f}\rangle\}$,
   $\mathtt{v}(q) \prec_k \Delta_q, \mathtt{v}(q) := \Delta_q, \mathtt{A} := \{q\}$
5. $p_i := q, \mathtt{A} := \emptyset, \mathtt{supp} := \{\langle q(a), \mathtt{f}\rangle, \langle r(b), \mathtt{f}\rangle\}$,
   $\mathtt{v}' := \{\langle q(a), \mathtt{f}\rangle, \langle q(b), \mathtt{t}\rangle, \langle r(a), \mathtt{t}\rangle, \langle r(b), \mathtt{f}\rangle\}, \Delta_q = \mathtt{v}(q)$
6. `stop. return` $\mathtt{v}(q)$

Iter i	Δ_{p_i}	$\mathtt{v}(p_i)$
0.	–	$\mathtt{v}(p_i) = \emptyset$
1.	$\Delta_q = \{\langle b, \mathtt{t}\rangle\}$	$\mathtt{v}(q) = \Delta_q$
2.	$\Delta_q = \{\langle b, \mathtt{t}\rangle\}$	–
3.	$\Delta_r = \{\langle a, \mathtt{t}\rangle, \langle b, \mathtt{f}\rangle\}$	$\mathtt{v}(r) = \Delta_r$
4.	$\Delta_q = \{\langle a, \mathtt{f}\rangle, \langle b, \mathtt{t}\rangle\}$	$\mathtt{v}(q) = \Delta_q$
5.	$\Delta_q = \{\langle a, \mathtt{f}\rangle, \langle b, \mathtt{t}\rangle\}$	–

Therefore, $Answer_{HF}(\mathcal{P}, \{q\}, H_2)$ returns $\Delta_q = \{\langle a, \mathtt{f}\rangle, \langle b, \mathtt{t}\rangle\}$ as expected.

It can then be shown that:

Proposition 4. *There is a limit ordinal λ such that after $|\lambda|$ steps $Answer_{HF}(\mathcal{P}, Q, H)$ returns the set of all correct answers of $\mathcal{P}$ with respect to the predicates in Q and the H-founded semantics.*

Termination of the query answering procedure is guaranteed whenever the termination of the basic procedure in Figure 2 is guaranteed (e.g., the same options as for the Kripke-Kleene semantics can be applied).

5 Conclusions

We have presented a simple, general, yet effective top-down algorithm to retrieve *all* correct answers to queries for normal logic programs under the AWA and, thus, under the CWA and OWA. To the best of our knowledge, this is the first time the problem of computing all answers has been addressed in this context, and under the CWA in particular, where arbitrary monotone functions in the body can manipulate truth values taken from a bilattice. We believe that its interest relies on its easiness for an effective implementation. Computing all answers is the first step towards top-k query answering, as it is developed in the context relational databases [9,15] and will be our primary topic of future research.

References

1. Alcantâra, J., Damásio, C.V., Pereira, L.M.: Paraconsistent logic programs. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, pp. 345–356. Springer, Heidelberg (2002)
2. Alferes, J.J., Pereira, L.M.: On logic program semantics with two kinds of negation. In: Proceedings of the Joint International Conference and Symposium on Logic Programming, pp. 574–588. The MIT Press, Washington (1992)
3. Arieli, O.: Paraconsistent declarative semantics for extended logic programs. Annals of Mathematics and Artificial Intelligence 36(4), 381–417 (2002)
4. Damásio, C.V., Pereira, L.M.: A survey of paraconsistent semantics for logic programs. In: Gabbay, D., Smets, P. (eds.) Handbook of Defeasible Reasoning and Uncertainty Management Systems, pp. 241–320. Kluwer, Dordrecht (1998)
5. Damásio, C.V., Pereira, L.M.: Antitonic logic programs. In: Eiter, T., Faber, W., Truszczyński, M. (eds.) LPNMR 2001. LNCS (LNAI), vol. 2173, Springer, Heidelberg (2001)
6. Dong, G., Libkin, L., Wong, L.: Incremental recomputation in local languages. Inf. Comput. 181(2), 88–98 (2003)
7. Dong, G., Su, J., Topor, R.W.: Nonrecursive incremental evaluation of datalog queries. Annals of Mathematics and Artificial Intelligence 14(2-4), 187–223 (1995)
8. Doyle, J., McDermott, D.: Nonmonotonic logic I. Artificial Intelligence 13, 41–72 (1980)
9. Fagin, R.: Combining fuzzy information: an overview. SIGMOD Rec. 31(2), 109–118 (2002)
10. Fitting, M.C.: Fixpoint semantics for logic programming - a survey. Theoretical Computer Science 21(3), 25–51 (2002)
11. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Kowalski, R.A., Bowen, K. (eds.) Proceedings of the 5th International Conference on Logic Programming, pp. 1070–1080. The MIT Press, Cambridge (1988)
12. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Computing 9(3/4), 365–386 (1991)
13. Ginsberg, M.L.: Multi-valued logics: a uniform approach to reasoning in artificial intelligence. Computational Intelligence 4, 265–316 (1988)
14. Leone, N., Rullo, P., Scarcello, F.: Disjunctive stable models: Unfounded sets, fixpoint semantics, and computation. Information and Computation 135(2), 69–112 (1997)
15. Li, C., Chang, K.C.-C., Ilyas, I.F., Song, S.: RankSQL: query algebra and optimization for relational top-k queries. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (SIGMOD-05), pp. 131–142. ACM Press, New York (2005)

16. Loyer, Y., Straccia, U.: Any-world assumptions in logic programming. Theoretical Computer Science 342(2-3), 351–381 (2005)
17. Loyer, Y., Straccia, U.: Epistemic foundation of stable model semantics. Journal of Theory and Practice of Logic Programming 6, 355–393 (2006)
18. Marek, V.W., Truszczyński, M.: Autoepistemic logic. Journal of the ACM 38(3), 587–618 (1991)
19. McCarthy, J.: Applications of circumscription to formalizing commonsense knowledge. Artificial Intelligence 28, 89–116 (1986)
20. McDermott, D.: Nonmonotonic logic II. journal of the ACM 22, 33–57 (1982)
21. Moore, R.C.: Semantical considerations on nonmonotonic logic. Artificial Intelligence 25, 75–94 (1985)
22. Reiter, R.: A logic for default reasoning. Artificial Intelligence 13, 81–132 (1980)
23. Straccia, U.: Query answering in normal logic programs under uncertainty. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 687–700. Springer, Heidelberg (2005)
24. Straccia, U.: Query answering under the any-world assumption for normal logic programs. In: Proceedings of the 10th International Conference on Principles of Knowledge Representation (KR-06), pp. 329–339. AAAI Press, Stanford (2006)
25. Ullman, J.D.: Principles of Database and Knowledge Base Systems, vol. 1,2. Computer Science Press, Potomac, Maryland (1989)
26. van Gelder, A., Ross, K.A., Schlimpf, J.S.: The well-founded semantics for general logic programs. Journal of the ACM 38(3), 620–650 (1991)
27. Vojtáš, P.: Fuzzy logic programming. Fuzzy Sets and Systems 124, 361–370 (2001)

Measure Logic*

Nebojša Ikodinović[2], Miodrag Rašković[1],
Zoran Marković[1], and Zoran Ognjanović[1]

[1] Mathematical Institute, Kneza Mihaila 35, 11000 Belgrade, Serbia
zoranm@mi.sanu.ac.yu, zorano@mi.sanu.ac.yu
[2] Prirodno-matematički fakultet, R. Domanovića 12, 34000 Kragujevac, Serbia
ikodinovic@kg.ac.yu

Abstract. In this paper we investigate logic which is suitable for reasoning about uncertainty in different situations. A possible-world approach is used to provide semantics to formulas. Axiomatic system for our logic is given and the corresponding strong completeness theorem is proved. Relationships to other systems are discussed.

1 Introduction

This paper is partly inspired by J. M. Keynes' work entitled "A Treatise on Probability" [10]. In this work, Keynes treats measuring and comparison of probabilities. He sharply criticizes the thesis that in principle every degree of probability can be measured, and argues that we are often not able to do that.

Keynes gives two sorts of examples. The first sort (in fact, very close to the approach in [8]): when, although we can say that the probability that A occurs is higher than the probability that B occurs, we still can not say how much is that probability higher (two, three times). And the second sort: when probabilities can not be compared at all. As illustration for impossibility of comparison of probabilities we can give the following example: "Suppose that there are three sets of experiments whose aim is to express a generalization. The first set of experiments is the most numerous, in the second set, the irrelevant conditions were more carefully modified, and in the third case, the generalization is larger than in the first two sets. Which of these generalizations is the most probable? There is no answer."

After analyzing Keynes' demands that a system of probabilities should respect the partial order among them, we came to the conclusion that it could be a lattice with the smallest element 0 (impossibility) and the biggest one 1 (certainty). We also consider it significant to keep the concept of (finite) additivity, although it does not follow from the Keynes' concept. We arrive thus to our operators $M_{=a}$, where a is an element of an ordered monoid.

One can argue that uncertainty should be measured by elements of more abstract systems (algebras) rather than by real numbers (the unit interval $[0, 1]$).

* This research was supported by Ministarstvo za nauku, tehnologije i razvoj Republike Srbije, through Matematički institut.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 128–138, 2007.

Such approach is taken in [7] where the unit interval is replaced by arbitrary poset. Semantically, our approach is similar, except that we add a binary operation, with the structure of commutative monoid, to the partially ordered set which is the range. Thus, our approach is mainly based on a binary operation unlike the one in [7] which is based on a binary relation. We need the binary operation in ordered to be able to express the (finite) additivity axiom. Main difference is that [7] does not have any syntax, as their system is intended as a semantics for default reasoning, while we introduce a logic for which our semantic is sound and complete.

We consider measure logic, denoted by L_{PG^+}, suitable for reasoning about (generalized) measure. The language is obtained by adding measure operators to classical propositional language. It allows making formulas such as $M_{=a}\alpha$, with the intended meaning "the measure of α is a". The measure operators behave like modal operators. As the corresponding semantics we introduce special types of Kripke models with addition of measure defined over the sets of worlds. Our emphasis is on complete axiomatizations of the logic. Following [12,14,18,21], we propose (infinitary) axiomatic system for which we prove the strong completeness theorem ('Every consistent set of formulas is satisfiable').

The rest of the paper is organized as follows. In Section 2 we define some basic notions and give a motivating example. In Section 3 the logic L_{PG^+} is introduced, and its syntax, semantics and axiomatization are given. We prove the corresponding strong completeness theorem. We conclude in Section 4 with some suggestions for further research.

2 Preliminaries

Any system designed for reasoning about the real world must be capable of dealing with uncertain information. From a logical point of view, uncertainty basically concerns formulas that can be either true or false, but their truth-value is unknown due to incompleteness of the available information. Among the different models of uncertainty, probability-like measure is one of the most relevant. Namely, it is most often supposed that uncertainty is a measurable feature, i.e. that uncertainty of an event can be expressed by the elements of a set G which is partially ordered by $\leq$ and which possesses additional algebraic structure $*, 0, \ldots$ So, measure of uncertainty is seen as a function $\mu : \mathcal{F} \to G$, where $\mathcal{F}$ is a given set of events. In our approach, the concept of event is classical: $\mathcal{F}$ is a Boolean algebra. The assumption of ordering of the range G is necessary for comparing degrees of uncertainty, i. e. for the formalization of the notions 'more uncertain than' and 'less uncertain than', etc., while the algebraic structure is necessary for calculating uncertainty degrees of complex events. The usual request is

$$\text{if } A \cap B = \emptyset, \text{ then } \mu(A \cup B) = \mu(A) * \mu(B),$$

where for the binary operation $*$ – so called *composition law* –: commutativity, associativity, monotonicity in both components, neutral element (and continuity) are required.

Definition 1. *Let* $\mathbf{G} = (G, \leq, *, 0)$ *be a partially ordered countable commutative monoid and* $G^+ = \{x \in G \mid 0 \leq x\}$. *A function* $\mu : \mathcal{F} \to G^+$ *is called* G^+*-measure on* $\mathcal{F}$ *if the following conditions are satisfied:*

1. $\mu(\emptyset) = 0$,
2. *if* $A, B \in \mathcal{F}$ *and* $A \cap B = \emptyset$, *then* $\mu(A \cup B) = \mu(A) * \mu(B)$.

There are a number of structures whose subsets can be very suitable for measuring of uncertainty.

1. additive monoid of nonnegative rational numbers $(Q^+, \leq, +, 0)$;
2. $(\{0, \frac{1}{n}, \ldots, 1\}, \leq, \oplus, 0)$, $x \oplus y = \min\{1, x + y\}$;
3. $([0,1]_Q, \leq, \oplus, 0)$ ($[0,1]_Q$ is the set of rational numbers from unit real interval $[0,1]$;
4. $(Q^+(\varepsilon), \leq, +, 0)$, where $Q^+(\varepsilon)$ is the set of nonnegative elements of the domain of the nonarchimedean field $\mathbf{Q}(\varepsilon)$ which is the smallest field obtained by adding positive infinitesimal ε to rational numbers.
5. $(Q^n, \leq, +, (0, \ldots, 0))$, $Q^n = \underbrace{Q \times \ldots Q}_{n}$, $\leq$ is lexicographic order;
6. $(\omega, \leq, +, 0)$;
7. $(\omega, \leq, \max, 0)$;
8. $(Q^+, \leq, \max, 0)$;

So called pseudo-additive measures have been studied and applied (Bacceli et al. (1992), Maslov et al. (1992), Pap (1995), Sugeno et al. (1987)). For the range of these measures is taken a semiring on the real interval $[a, b]$, with the corresponding operations pseudo-addition and pseudo-multiplication. A great number of examples of these measures is given in [17]. We emphasise that these measures can be applied in very different situations (system theory, optimization, control theory, differential equations, difference equations, etc.).

Example 1. For the weather forcast, we can ask the question: "Is the probability that on Saturday will be a nice day higher than the probability that on Friday will be a nice day?" Naturally, in order to answer this question we should first define what means the rather vague notion of "nice day". If we suppose that nice day is when the temperature is between 15° and 25°, humidity 40%, the wind slower than 3 m/s etc., then the probability of 'nice day' is presented by the (finite) sequence of probabilities that each relevant parameter takes a value in corresponding interval. Therefore, it make sense to say that the probability of 'nice day on Saturday' is higher than the probability of 'nice day on Friday', if each coordinate in probability for Saturday is higher or equal to the corresponding one for Friday, in other cases the probabilities can not be compared.

3 The Logic L_{PG^+}

3.1 Syntax

Let G stand for denumerable partially ordered commutative monoid.

The language of L_{PG^+} consists of a countable set $I = \{p_1, p_2, \ldots\}$ of propositional letters, classical connectives $\wedge$ and $\neg$, and a list of unary operators $M_{=a}$ for every $a \in G^+$.

The set $For^C(L_{PG^+})$ of all classical propositional formulas is defined inductively as the smallest set X containing propositional letters and closed under the usual formation rules: if α and β belong to X, then $\neg\alpha$, $\alpha \wedge \beta$ are in X. Elements of $For^C(L_{PG^+})$ will be denoted by $\alpha, \beta, \ldots$ The set $For^M(L_{PG^+})$ of all measure formulas is the smallest set Y containing all formulas of the form: $M_{=a}\alpha$, for each $\alpha \in For^C(L_{PG^+})$ and each $a \in G^+$, and closed under the formation rules: if A and B belong to Y, then $\neg A$ and $A \wedge B$ are in Y. The formulas from $For^M(L_{PG^+})$ will be denoted by A, B, ... Let $For(L_{PG^+}) = For^C(L_{PG^+}) \cup For^M(L_{PG^+})$. The formulas from $For(LPP)$ will be denoted by Φ, Ψ, ...

Note that neither mixing of pure propositional formulas and measure formulas, nor nested measure operators are allowed.

We use the usual abbreviations for the other classical connectives $\vee$, $\rightarrow$, $\leftrightarrow$. Also, for every a from G^+ we denote $\neg M_{=a}(\alpha)$ by $M_{\neq a}(\alpha)$

For $\alpha \in For^C(L_{PG^+})$, and $A \in For^M(L_{PG^+})$, we abbreviate both $\alpha \wedge \neg\alpha$ and $A \wedge \neg A$ by $\bot$ letting the context determine the meaning, while $\top$ denotes $\neg\bot$.

3.2 Semantics

The semantics for $For(L_{PG^+})$ will be based on the possible-world approach.

Definition 2. *An $L_{PG^+}-$model is a structure $\mathbf{M} = \langle W, \mathcal{F}, \mu, v\rangle$ where:*

- *W is a nonempty set of objects called worlds,*
- *$\mathcal{F}$ is an algebra of subsets of W,*
- *μ is a G^+-measure on $\mathcal{F}$,*
- *$v : W \times I \rightarrow \{true, false\}$ provides for each world $w \in W$ a two-valued evaluation of the propositional letters, that is $v(w, p) \in \{true, false\}$, for each propositional letter $p \in I$ and each world $w \in W$; a truth-evaluation $v(w, \cdot)$ is extended to classical propositional formulas as usual.*

If $\mathbf{M}$ is an $L_{PG^+}-$model and $\alpha \in For^C(L_{PG^+})$, the set $\{w : v(w, \alpha) = true\}$ is denoted by $[\alpha]_{\mathbf{M}}$. We will omit the subscript $\mathbf{M}$ from $[\alpha]_{\mathbf{M}}$ and write $[\alpha]$ if $\mathbf{M}$ is clear from the context. An $L_{PG^+}-$model $\mathbf{M} = \langle W, \mathcal{F}, \mu, v\rangle$ is measurable if $[\alpha]_{\mathbf{M}} \in \mathcal{F}$ for every formula $\alpha \in For^C(L_{PG^+})$. In this section we focus on the class of all measurable models (denoted by $\mathrm{Meas}(L_{PG^+})$).

Definition 3. *The satisfiability relation is defined inductively by the following conditions, for every $L_{PG^+}-$model $\mathbf{M} = \langle W, \mathcal{F}, \mu, v\rangle$:*

– *if* $\alpha \in For^C(L_{PG^+})$, $\mathbf{M} \models \alpha$ *iff for every world* $w \in W$, $v(w, \alpha) = true$,
– *if* $\alpha \in For^C(L_{PG^+})$, $\mathbf{M} \models M_{=a}\alpha$ *iff* $\mu([\alpha]) = a$,
– *if* $A \in For^M(L_{PG^+})$, $\mathbf{M} \models \neg A$ *iff* $\mathbf{M} \not\models A$,
– *if* $A, B \in For^M(LPP)$, $\mathbf{M} \models A \wedge B$ *iff* $\mathbf{M} \models A$ *and* $\mathbf{M} \models B$.

A formula $\Phi \in For(L_{PG^+})$ *is satisfiable if there is a measurable* L_{PG^+}*-model* $\mathbf{M} = \langle W, \mathcal{F}, \mu, v\rangle$ *such that* $\mathbf{M} \models \Phi$*;* Φ *is valid if for every measurable* L_{PG^+}*-model* $\mathbf{M}$, $\mathbf{M} \models \Phi$*; a set of formulas is satisfiable if there is a model in which every formula from the set is satisfiable.*

3.3 Axiomatic System

The axiomatic system $Ax(L_{PG^+})$ for L_{PG^+} contains the following axiom schemata:

1. all $For^C(L_{PG^+})-$instances of classical propositional tautologies,
2. all $For^M(L_{PG^+})-$instances of classical propositional tautologies,
3. $M_{=a}\alpha \rightarrow \neg M_{=b}\alpha$, $a \neq b$,
4. $M_{=0}\neg(\alpha \leftrightarrow \beta) \rightarrow (M_{=a}\alpha \rightarrow M_{=a}\beta)$,
5. $(M_{=a}\alpha \wedge M_{=b}\beta \wedge M_{=0}(\alpha \wedge \beta)) \rightarrow M_{=a*b}(\alpha \vee \beta)$

and inference rules:

1. From Φ and $\Phi \rightarrow \Psi$ infer Ψ,
2. From α infer $M_{=0}\neg\alpha$,
3. From $A \rightarrow M_{\neq a}\alpha$, for every $a \in G^+$, infer $\neg A$.

Note that the inference rule 3 is infinitary if and only if G is an infinite set. By this rule, at the syntax level, we define the range of the measure.

Definition 4. *A formula* Φ *is deducible from a set* T *of formulas* ($T \vdash \Phi$) *if there is an at most countable sequence of formulas* $\Phi_0, \Phi_1, \ldots, \Phi$*, such that every* Φ_i *is an axiom or a formula from the set* T*, or it is derived from the preceding formulas by an inference rule.*

A formula Φ *is a theorem* ($\vdash \Phi$) *if it is deducible from the empty set, and a proof for* α *is the corresponding sequence of formulas.*

A set T *of formulas is consistent if there is at least one formula from* $For^C(L_{PG^+})$*, and at least one formula from* $For^P(L_{PG^+})$ *that are not deducible from* T*, otherwise* T *is inconsistent.*

A consistent set T *of formulas is said to be maximal consistent if the following holds:*

– *for every* $\alpha \in For^C(L_{PG^+})$*, if* $T \vdash \alpha$*, then* $\alpha \in T$ *and* $M_{=0}\neg\alpha \in T$*, and*
– *for every* $A \in For^P(L_{PG^+})$*, either* $A \in T$ *or* $\neg A \in T$.

A set T *is deductively closed if for every* $\Phi \in For(L_{PG^+})$*, if* $T \vdash \Phi$*, then* $\Phi \in T$.

3.4 Soundness and Completeness

Theorem 1 (Soundness theorem). *The axiomatic system $Ax(L_{PG^+})$ is sound with respect to the class of measurable L_{PG^+}-models.*

Proof. Soundness of our system follows from the soundness of propositional classical logic and from the properties of G^+-measure.We can show that every instance of an axiom schema holds in every measurable L_{PG^+}-model, while the inference rules preserve the validity.

Let $\mathbf{M} = \langle W, \mathcal{F}, \mu, v\rangle$ be an arbitrary measurable L_{PG^+}-model.

It is easy to see that if Φ is an instance of a classical propositional tautology, then $\mathbf{M} \models \Phi$. Axioms 4 and 5 concern the properties of G^+-measures and obviously holds in every model. For example, let us consider Axiom 4. If $\mathbf{M} \models M_{=0}\neg(\alpha \leftrightarrow \beta)$ and $\mathbf{M} \models M_{=a}\alpha$, then $\mu[\neg(\alpha \leftrightarrow \beta)] = 0$ and $\mu[\alpha] = a$. From $[\neg(\alpha \leftrightarrow \beta)] = ([\neg\alpha] \cap [\beta]) \cup ([\alpha] \cap [\neg\beta])$ and $([\neg\alpha] \cap [\beta]) \cap ([\alpha] \cap [\neg\beta]) = \emptyset$ we have

$$0 = \mu[\neg(\alpha \leftrightarrow \beta)] = \mu([\neg\alpha] \cap [\beta]) * \mu([\alpha] \cap [\neg\beta]),$$

So, it follows that $\mu([\neg\alpha] \cap [\beta]) = 0$ and $\mu([\alpha] \cap [\neg\beta]) = 0$. Now, we have

$$\mu[\beta] = \mu([\alpha] \cap [\beta]) * \mu([\neg\alpha] \cap [\beta]) = \mu([\alpha] \cap [\beta])$$

and

$$\mu[\alpha] = \mu([\alpha] \cap [\beta]) * \mu([\alpha] \cap [\neg\beta]) = \mu([\alpha] \cap [\beta]).$$

Hence, $\mu[\beta] = \mu[\alpha] = a$, and $\mathbf{M} \models M_{=a}\beta$.

Consider Rule 3. Suppose that $\mathbf{M} \models A \rightarrow M_{\neq a}\alpha$, for every $a \in G^+$. If $\mathbf{M} \not\models \neg A$, then $\mathbf{M} \models A$ and $\mathbf{M} \models M_{\neq a}\alpha$, for each $a \in G^+$, i.e. $\mu([\alpha]) \neq a$, for each $a \in G^+$, which is a contradiction. □

It is easy to see that the following inference rule

$$\text{from } \alpha \leftrightarrow \beta, \text{ infer } M_{=a}\alpha \leftrightarrow M_{=a}\beta, \text{ for every } a \in G^+,$$

is derivable.

In order to prove the completeness theorem for our logic, we follow the Henkin style procedure. We begin with some set of statements. Then, we describe how a consistent set T of formulas can be extended to a suitable maximal consistent set, and how a canonical model can be constructed out of such maximal consistent sets.

Theorem 2
(Deduction theorem) *If T is a set of formulas, Φ is a formula, and $T \cup \{\Phi\} \vdash \Psi$, then $T \vdash \Phi \rightarrow \Psi$, where Φ and Ψ are either both classical or both measure formulas.*

Proof. We use the transfinite induction on the length of the proof of Ψ from $T \cup \{\Phi\}$. The classical cases follow as usual. Suppose $\Psi = \neg A$ is obtained from $T \cup \{\Phi\}$ by an application of Rule 3 with premises $A \rightarrow M_{\neq a}\alpha$, for each $a \in G^+$. In that case, we have:

$T \cup \{\Phi\} \vdash A \to M_{\neq a}\alpha$, for each $a \in G^+$,
$T \vdash \Phi \to (A \to M_{\neq a}\alpha)$, for each $a \in G^+$, by the induction hypothesis
$T \vdash (\Phi \wedge A) \to M_{\neq a}\alpha$, for each $a \in G^+$,
$T \vdash \neg(\Phi \wedge A)$, by Rule 3
$T \vdash \Phi \to \neg A$
$T \vdash \Phi \to \Psi$. □

Theorem 3. *Every consistent set of formulas can be extended to a maximal consistent set.*

Proof. Let T be a consistent set, $Con^C(T)$ the set of all classical formulas that are consequences of T, A_0, A_1, ..., an enumeration of $For^P(L_{PG^+})$, and α_0, α_1, ..., an enumeration of $For^C(L_{PG^+})$. We define a sequence of sets T_i, $i = 0, 1, 2, \ldots$, such that:

1. $T_0 = T \cup Con^C(T) \cup \{M_{=0}\neg\alpha : \alpha \in Con^C(T)\}$,
2. for every $i \geq 0$, if $T_{2i} \cup \{A_i\}$ is consistent, then $T_{2i+1} = T_{2i} \cup \{A_i\}$; otherwise $T_{2i+1} = T_{2i} \cup \{\neg A_i\}$,
3. for every $i \geq 0$, $T_{2i+2} = T_{2i+1} \cup \{M_{=a}\alpha_i\}$, for some $a \in G^+$, so that T_{2i+2} is consistent.

T_0 is consistent because it is a set of consequences of a consistent set. Suppose that T_{2i+1} is obtained by the step 2 of the above construction and that neither $T_{2i} \cup \{A_i\}$, nor $T_{2i} \cup \{\neg A_i\}$ are consistent. It follows by the deduction theorem that $T_{2i} \vdash A_i \wedge \neg A_i$, which is a contradiction. Consider the step 3. If $T_{2i+1} \cup \{M_{=a}\alpha_i\}$ is not consistent for every $a \in G^+$, then:
$T_{2i+1} \vdash M_{=a}\alpha \to \bot$, $a \in G^+$
$T_{2i+1} \vdash \top \to M_{\neq a}\alpha$, $a \in G^+$
$T_{2i+1} \vdash \neg\top$ (by Rule 3)
i.e. T_{2i+1} is not consistent. Hence, it is always possible to produced the consistent extension by the step 3.

We continue by showing that the set $\overline{T} = \cup_{i \geq 0} T_i$ is a deductively closed set which does not contain all formulas, and, as a consequence, that $\overline{T}$ is consistent. If $\alpha \in For^C(L_{PG^+})$, by the construction of T_0, α and $\neg\alpha$ cannot be simultaneously in T_0. For a formula $A \in For^P(L_{PG^+})$, either $A \in \overline{T}$ or $\neg A \in \overline{T}$, and the set $\overline{T}$ does not contain both $A = A_i$ and $\neg A = A_j$, because $T_{\max\{i,j\}+1}$ is a consistent set. Next, note that for every $\Phi \in For^P(L_{PG^+})$, if $T_i \vdash \Phi$, then it must be $\Phi \in \overline{T}$. Namely, if $T_i \vdash A_k$, for some $i \geq 0$, and $A_k \notin \overline{T}$, then $\neg A_k \in \overline{T}$, and $T_{\max\{i,2k+1\}+1}$ is not consistent, which is a contradiction.

Suppose that $\overline{T} \vdash \Phi$. If $\Phi \in For^C(L_{PG^+})$, then by the construction of T_0, $\Phi, M_{=0}\neg\Phi \in \overline{T}$. Let $\Phi \in For^P(L_{PG^+})$. Suppose that the sequence $\Phi_1, \Phi_2, \ldots, \Phi$ of formulas which forms the proof of Φ from $\overline{T}$ is countably infinite (otherwise there must be some k such that $T_k \vdash \Phi$, and it follows that $\Phi \in \overline{T}$). We can show that for every i, if Φ_i is obtained by an application of an inference rule, and all the premises of Φ_i belong to $\overline{T}$, then $\Phi_i \in \overline{T}$. Suppose Φ_i is obtained by Rule 1 and its premises Φ_i^1 and Φ_i^2 belong to $\overline{T}$. There must be some k such that $\Phi_i^1, \Phi_i^2 \in T_k$. Since $T_k \vdash \Phi_i$, it must be $\Phi_i \in \overline{T}$. If Φ_i is obtained by Rule 2, its premise

belongs to T_0 and the same holds for Φ_i. Suppose that $\Phi_i = \neg A$ is obtained by the infinitary inference rule 3, and that the premises $\Phi_i^a = A \to M_{\neq a}\alpha$, $a \in G^+$, belong to $\overline{T}$. Suppose $\Phi_k \notin \overline{T}$. Then $A \in \overline{T}$, and $A \in T_i$, for some $i \geq 0$. On the other hand, if $\alpha = \alpha_j$, then, by the step 3, for some $a_j \in G^+$, $M_{=a_j}\alpha_j \in T_{2j+2}$. Also, there is an index m so that $\Phi_i^{a_j} = A \to M_{\neq a_j}\alpha_j \in T_m$. Hence, $A, M_{=a_j}\alpha_j, A \to M_{\neq a_j}\alpha_j \in T_{\max\{i,2j+2,m\}+1}$, which is a contradiction.

So, from $\overline{T} \vdash \Phi$, it follows $\Phi \in \overline{T}$, i.e., the set $\overline{T}$ is deductively closed. Since it does not contain all formulas, $\overline{T}$ is consistent, while the construction guarantees that it is maximal. □

Theorem 4 (Completeness theorem). *Every consistent set T of formulas has a measurable L_{PG^+}-model.*

Proof. Using the maximal consistent extension $\overline{T}$ of the set T, we can define a tuple $\mathbf{M} = \langle W, \mathcal{F}, \mu, v\rangle$, where:

- W contains all the classical propositional interpretations that satisfy the set $Con^C(T)$ of all classical consequence of the set T, i.e. $W = \{w : w \models Con^C(T)\}$,
- for every $\alpha \in For^C(L_{PG^+})$, $[\alpha] = \{w \in W : w \models \alpha\}$ and $\mathcal{F} = \{[\alpha] : \alpha \in For^C(L_{PG^+})\}$,
- $v : W \times I \to \{true, false\}$ is an assignment such that for every world $w \in W$ and every propositional letter $p \in I$, $v(w, p) = true$ iff $w \models p$,
- $\mu : \mathcal{F} \to G^+$, such that $\mu([\alpha]) = a$ iff $M_{=a}\alpha \in \overline{T}$.

Note that, since w's are classical propositional interpretations, in the above definition of $\mathbf{M}$ we use $w \models \alpha$ to denote that the interpretation w satisfies α in the sense of classical propositional logic.

First we have to prove that $\mathbf{M}$ is a measurable L_{PG^+}-model. $\mathcal{F}$ is an algebra of subsets of W. Indeed, for an arbitrary formula α, $W = [\alpha \vee \neg\alpha] \in \mathcal{F}$. Also, if $[\alpha] \in \mathcal{F}$, then the complement of $[\alpha]$ is the set $[\neg\alpha]$, and it belongs to $\mathcal{F}$, and if $[\alpha_1], \ldots [\alpha_n] \in \mathcal{F}$, then the union $[\alpha_1] \cup \ldots \cup [\alpha_n] \in \mathcal{F}$ because $[\alpha_1] \cup \ldots \cup [\alpha_n] = [\alpha_1 \vee \ldots \vee \alpha_n]$. Thus, $\mathcal{F}$ is an algebra of subsets of W. Next, we will show that μ is an G^+-measure on $\mathcal{F}$. If $[\alpha] = [\beta]$, then, by the completeness theorem for classical classical propositional logic, $Con^C(L_{PG^+}) \vdash \alpha \leftrightarrow \beta$, and $M_{=0}\neg\alpha \leftrightarrow \beta \in \overline{T}$, by Rule 3. From Axiom 4, it follows that μ is well-defined. Let $[\alpha] \cap [\beta] = \emptyset$, $M_{=a}\alpha, M_{=b}\beta \in \overline{T}$. By the completeness theorem for classical classical propositional logic and Rule 3, we have $\overline{T} \vdash M_{=0}(\alpha \wedge \beta)$. Thus, by Axiom 5, it follows $M_{=a*b}(\alpha \vee \beta) \in \overline{T}$, i.e. $\mu([\alpha] \cup [\beta]) = a * b$. Hence, $\mathbf{M}$ is a measurable L_{PG^+}-model.

Finally, we can show that for every formula Φ, $\mathbf{M} \models \Phi$ iff $\Phi \in \overline{T}$, and consequently that the set T is satisfiable. If Φ is a classical formula, the statement obviously holds. If $\Phi = M_{=a}\alpha$, then

$$\mathbf{M} \models M_{=a}\alpha \text{ iff } \mu([\alpha]) = a \text{ iff } M_{=a} \in \overline{T}. \qquad \square$$

4 Conclusion

We have presented a measure logic which is suitable for reasoning about different models of uncertainty. This approach allows to extend standard probabilistic reasoning when we consider different additive structures (e.g. examples from section 2). Nonarchimedean additive monoids correspond to the first kind of Keynes' remarks where two probabilities are comparable but still it is not possible to say how many times one is higher than another. The second kind of Keynes' remarks (un-comparable probabilities) can be treated using partially ordered monoids. We have given a sound and complete axiomatic system with infinitary rules of inferences. This framework is very general and can be used for other similar logics. For example, in a similar way we could extend propositional inuitionistic logic. Namely, one could introduce the notion of an G^+-measure in exactly the same way as above on general domains as rings of sets.

It would be very useful to study the natural connections between L_{PG^+} and many other logics: temporal logics, possibility logic, conditional logic, multivalued logics, fuzzy logic, etc:

- If G is a set of sequences of rational numbers, and in $M_{=a}\alpha$ the sequence a denotes values of the probability of α along a discrete linear-time line, we can enrich probabilistic reasoning with some temporal features.
- If the range of measure is the domain of the monoid $(\{0,1,2,\ldots\}\cup\{\infty\},\min,\infty)$, so that $\mu(W)=0$ and $\mu(\emptyset)=\infty$, we obtain a plausibility space known as ordinal ranking [8].
- If the range of measure is the unit interval $[0,1]$ and we choose $*$ to be max (sup), i.e. $\mu(W)=1$, $\mu(\emptyset)=0$ and $\mu(A)=\sup_{w\in A}(\mu(w))$, we obtain the structures suitable for semantical characterization of possibility logic [3,4].
- If the range of measure is the unit interval $[0,1]$ provided with the binary operation $(a,b)\mapsto a+b+\lambda ab$, where $\lambda\in(-1,+\infty)$, we obtain the well-known fuzzy Sugeno $\lambda-$measure.
- The disjunctive fragment of fuzzy logic can be modelled if we choose $*$ to be one of the well-known t-conorms [2,4,17,24].

As we stressed above, by suitable choice of the range G the logic L_{PG^+} can be adapted to many real situations which require reasoning about uncertainty. Thus, the further investigations of the logic can be developed in various directions, both practical and theoretical. It would be very interesting to develop procedures to compare logics $L_{PG_1^+}$ and $L_{PG_2^+}$, for different ranges G_1 and G_2.

An important example of logic L_{PG^+} corresponds to G^+-measure μ on a field $\mathcal{F}$ of subsets of Ω, where $(G,\leq,*,0)$ is a *linearly* ordered commutative monoid. In that case we can introduce 'weaker' operators: $M_{\geq a}$ and $M_{\leq a}$, and then the operator $M_{=a}$ is definable by them: $M_{=a}\alpha \stackrel{\text{def}}{=} M_{\geq a}\alpha\wedge M_{\leq a}\alpha$. In a completely analog way, we can give the corresponding axiomatization. Logic L_{PG^+} can be also extended so that it can permit qualitative comparison of uncertainty, in the way similar to the one given in [15].

Finally, let us mention that decidability is very important for applicability of a logic system. As in many similar logics [12,14,18] the problem of decidability,

i. e. the problem of satisfiability of a formula is equivalent to satisfiability in G^+ of a of equalities and negations of equalities of the forms: $x_1 * x_2 * \ldots * x_m = a$ and $x_1 * x_2 * \ldots * x_m \neq b$, for some a and b from G^+. It is obvious that the logic L_{PG^+} is decidable if the set G is finite. Also, for some infinite sets G the corresponding logics are decidable. Determining more precisely for which infinite sets G this holds seems to be an interesting subject for further investigation.

References

1. Baccelli, F., Cohen, G., Olsder, G.J., Quadrat, J.P.: Synchronization and Linearity: An algebra for Discrete Evants System. Wiley, New York (1992)
2. Dubois, D., Prade, H.: A class of fuzzy measures based on triangular norms. A general framework for the combination of uncertain information, Int. J. of General System 8(1), 43–61 (1982)
3. Dubois, D., Prade, H.: Possibility Theory. Plenum Press, New York (1988)
4. Dubois, D., Prade, H.: An introduction to possibility and fuzzy logic. In: Shafer, G., Pearl, J. (eds.) Readings in Uncertainty Reasoning, pp. 742–761. Morgan Kaufmann, San Francisco (1990)
5. Fagin, R., Halpern, J., Megiddo, N.: A logic for reasoning about probabilities. Information and Computation 87(1/2), 78–128 (1990)
6. Fagin, R., Halpern, J.: Reasoning about knowledge and probability. Journal of the ACM 41(2), 340–367 (1994)
7. Friedman, N., Halpern, J.: Plausibility Measure and Default Reasoning. Journal of the ACM 48(4), 648–685 (2001)
8. Goldszmidt, M., Pearl, J.: Rank-based system: A simple approach to belief revision, belief update and reasoning about evidence and actions. In: Nebel, B., Rich, C., Swartout, W. (eds.) Proc. Third International Conference on Principle of Knowledge Representation and Reasoning (KR '92), pp. 661–672. Morgan Kaufmann, San Francisco (1992)
9. Hoover, D.N.: Probability logic. Annals of mathematical logic 14, 287–313 (1978)
10. Keynes, J.M.: A Treatise on Probability. MacMillan and Co., London (1943)
11. Lukasiewicz, T.: Probabilistic Default Reasoning with Conditional Constraints. Annals of Mathematics and Artificial Intelligence 34, 35–88 (2002)
12. Marković, Z., Ognjanović, Z., Rašković, M.: A Probabilistic Extension of Intuitionistic Logic. Mathematical Logic Quarterly 49(5), 415–424 (2003)
13. Maslov, V.P., Samborskii, S.N. (eds.): Idempotent Analysis, Adv. Sov. Math. Amer. Math. Soc., Providence, RI, vol. 13 (1992)
14. Ognjanović, Z., Rašković, M.: Some probability logics with new types of probability operators. Journal of logic and Computation 9(2), 181–195 (1999)
15. Ognjanović, Z., Perović, A., Rašković, M.: Logics with Qualitative Probability Operator (submitted)
16. Pap, E.: Null-Additive Set Function. Kluwer, Dordrecht (1995)
17. Pap, E. (ed.): Handbook of Measure Theory. Elseiver (2002)
18. Rašković, M.: Classical logic with some probability operators. Publication de l'Institut Math. (NS) 53(67), 1–3 (1993)
19. Rašković, M., Đorđević, R.: Probability quantifiers and operators, Vesta, Beograd (1996)
20. Rašković, M., Ognjanović, Z., Marković, Z.: A probabilistic Approach to Default Reasoning. In: 10th International Workshop on Non-Monotonic Reasoning NMR2004, June 6-8, 2004, Westin Whistler, Canada (2004)

21. Rašković, M., Ognjanović, Z., Marković, Z.: A Logic with Conditional Probabilities. In: Alferes, J.J., Leite, J.A. (eds.) JELIA 2004. LNCS (LNAI), vol. 3229, pp. 226–238. Springer, Heidelberg (2004)
22. Shafer, G.: A Mathematical Theory of Evidence. Princendon University Press, Princendon (1976)
23. Sugeno, M., Murofushi, T.: Pseudo-additive measures and integrals. J. Math. Anal. Appl. 122, 197–222 (1987)
24. Weber, S.: Decomposable measures and integrals for Archimedean t-conorms. J. Math. Anal. and Appl. 101(1), 114–138 (1984)

Weak Implication in Terms of Conditional Uncertainty Measures

Giulianella Coletti[1], Romano Scozzafava[2], and Barbara Vantaggi[2]

[1] Università di Perugia (Italy)
coletti@dipmat.unipg.it
[2] Università di Roma "La Sapienza" (Italy)
romscozz,vantaggi@dmmm.uniroma1.it

Abstract. We define weak implication $H \longmapsto_\varphi E$ ("H weakly implies E under φ") through the relation $\varphi(E|H) = 1$, where φ is a (coherent) conditional uncertainty measure. By considering various such measures with different levels of generality, we get different sets of "inferential rules", that correspond to those of default logic when φ reduces to a conditional probability.

Keywords: conditional uncertainty measures, weak implication, default logic.

1 Introduction

How can a seemingly loose concept like "weak implication" be embedded in a rigorous mathematical framework? There is a huge relevant literature on this matter, and lack of space does not allow us to undertake a discussion or a review on this. Just to recall some "semantic" aspects, let us consider the trivial statement "if it rains on the spot x, then x is wet"; this is clearly a logical implication $R \subseteq W$ (with obvious meaning of the symbols concerning the *events* R and W). Conversely, assuming W (x is wet), we could conclude R (it rains on the spot x) if we are not made aware of possible water sources around x: shortly, we may say "W weakly implies R" (or else that the "rule" $W \subseteq R$ rarely has exceptions).

Now, given a conditional (uncertainty) measure φ (for precise definitions, see Section 2), we can represent the above situation by the notation $W \longmapsto_\varphi R$ and by assessing $\varphi(R|W) = 1$, but for a rigorous formulation we need to say something more on events and conditional events.

An *event* can be singled-out by a (nonambiguous) statement E, that is a (Boolean) *proposition* that can be either *true* or *false* (corresponding to the two "values" 1 or 0 of the *indicator* I_E of E). The "logic of certainty" deals with *true* and *false* as final, and *not asserted*, answers concerning a *possible* event, while two particular cases are the *certain* event Ω (that is always true) and the *impossible* event $\emptyset$ (that is always false): notice that *only in these two particular cases the relevant propositions correspond to an assertion*. To make an assertion, we need to say something extra-logical, such as "You know that E is false" (so that $E = \emptyset$).

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 139–150, 2007.

In this paper we consider different classes of conditional measures φ, and on the basis of a suitable definition of weak implication we search for relevant "inferential rules", whose number should increase as the level of generality of the measure φ decreases. In particular, we show that when φ reduces to a (coherent) conditional probability P, we get exactly the rules for *default reasoning* (relevant references are [15] and [16]) as given, e.g., by Lehmann & Magidor [13]. In other words, in our framework default logic can be seen as a particular case of weak implication – in terms of φ – when we take as φ a coherent conditional probability.

To make clear our "strategy", let us refer to the more familiar case of probability. As it is well–known , a *certain* event (that is, an event *known to be true*) has probability 1, but *not* conversely (so that we may have many *possible* events whose probability is 1). Given a conditional event $E|H$, notice that $P(E) = 1$ does not imply $P(E|H) = 1$ (as in the usual framework where it is necessary to assume $P(H) > 0$). We can take instead $P(H) = 0$ (the *conditioning* event H – which *must* be a *possible* one – may in fact have *zero probability*, since in the assignment of $P(E|H)$ we are driven only by coherence [6]). Then a probability equal to 1 *can be, in our framework, updated*.

Moreover, $P(E|H) = 1$ does not imply $H \subseteq E$: take in fact, e.g., an event E with $P(E) > 0$ and an event $H \supset E$ such that $P(H) = P(E)$ (that is $P(E^c \wedge H) = 0$). In particular, if we *assert* $H^c \vee E = \Omega$, then $H \subseteq E$ (H logically implies E), so we certainly have $P(E|H) = 1$.

It could appear that the most "natural" way to weaken inclusion should be the requirement $P(H^c \vee E) = 1$. But this weaker assumption in general is not enough (whenever $P(H) = 0$) to get $P(E|H) = 1$, while $P(E|H) = 1$ always entails $P(H^c \vee E) = 1$. The two latter statements easily follow from

$$P(H^c \vee E) = 1 - P(H \wedge E^c) = 1 - P(H)P(E^c|H)\,.$$

In other (semantic) words, we require that, even if a part of H is not inside E, this part can be considered, in a sense, as "ignorable" (with respect to H itself), while the probability of $H^c \vee E$ can be equal to 1, due to the circumstance that H may have probability equal to 0 even if a "large part" of it is not inside E.

Other relevant aspects and results, based on the *truth values* of conditional events and on the ensuing logical relations involving unconditional ones, are discussed in the paper [14].

The previous considerations should make clear why we choose (given a suitable conditional measure φ) as definition of $H \longmapsto_\varphi E$ simply the equality $\varphi(E|H) = 1$.

2 Conditional Measures and Coherent Partial Assessments

We consider conditional measures $\varphi(\cdot|\cdot)$ through a *direct* introduction of a function whose domain is an arbitrary set of conditional events.

We recall that in this way we can define a conditional measure for any pair of events E, H, with $H \neq \emptyset$, and the knowledge (or the assessment) of the "joint" and "marginals" unconditional measures $\varphi(E \wedge H)$ and $\varphi(H)$ is not

required. Obviously, if the latter are already given, there must exist suitable rules that put them in relation with $\varphi(E|H)$, but the converse is not necessarily true. In particular, there is no need, as in the usual approaches – where the conditional measure is introduced by *definition* as a suitable function of the two aforementioned unconditional measures – of any specific assumption (such as – referring to the case of probability – the requirement of positivity for the measure of the given conditioning event).

Our interest is focused on conditional measures $\varphi(\cdot|\cdot)$ such that there exist two operations $\oplus$ and $\odot$ which render φ formally (or, better, essentially) "similar" to a conditional probability.

2.1 Decomposable Conditional Measures

Given a Boolean algebra $\mathcal{E}$, a function

$$\varphi : \mathcal{E} \to [0, 1]$$

is a $\oplus$-decomposable measure if $\varphi(\Omega) = 1, \varphi(\emptyset) = 0$ and there exists a commutative, associative and increasing operation $\oplus$ from $(\mathbb{R}_o^+)^2$ to $\mathbb{R}_o^+$, with 0 as neutral element and such that the following condition holds: for every $E_i, E_j \in \mathcal{E}$, with $E_i \wedge E_j = \emptyset$,

$$\varphi(E_i \vee E_j) = \varphi(E_i) \oplus \varphi(E_j). \tag{1}$$

Two particular well-known cases of decomposable measures are *probability* (where $\oplus$ is the standard sum) and *possibility* (where $\oplus$ is the maximum).

Definition 1. *Let φ be a real function defined on $\mathcal{C} = \mathcal{E} \times \mathcal{H}$, with $\mathcal{E}$ a Boolean algebra, $\mathcal{H} \subseteq \mathcal{E}$ an additive set (i.e., closed with respect to finite logical sums) with $\emptyset \notin \mathcal{H}$. Then the function φ is a $(\oplus, \odot)$–decomposable* conditional *measure if there exist two commutative, associative and increasing operations $\oplus, \odot$ from $(\mathbb{R}_o^+)^2$ to $\mathbb{R}_o^+$, having, respectively, 0 and 1 as neutral elements, and with $\odot$ distributive over $\oplus$, such that:*

(C1) $\varphi(E|H) = \varphi(E \wedge H|H)$, for every $E \in \mathcal{E}$ and $H \in \mathcal{H}$,

(C2) for any given $H \in \mathcal{H}$ the function $\varphi(\cdot|H)$ is a $\oplus$-decomposable measure,

(C3) for every $A \in \mathcal{E}$ and E, H, $E \wedge H \in \mathcal{H}$,

$$\varphi\big((E \wedge A)|H\big) = \varphi(E|H) \odot \varphi\big(A|(E \wedge H)\big).$$

Then different (decomposable) conditional measures can be obtained by particular choices of the two operations $\oplus$ and $\odot$. For example, choosing ordinary sum and product, or *max* and any t-norm, we get, respectively, conditional probability as given by de Finetti [11] and (in an equivalent form) by Popper [**?**] (see [5]), or conditional possibility (see [1]).

Note that *(C1)* and *(C2)* imply

(C1') $\varphi(H|H) = 1$, for every $H \in \mathcal{H}$.

2.2 Weakly and Generalized Decomposable Measures

We introduce different classes of measures singled-out on the basis of the set of events (conditional or not) where suitable properties of the operations $\oplus$ and $\odot$ are required to hold.

Even if the above definition of decomposable measure is very general, nevertheless it is unable to capture many well known ones, for example belief functions and convex capacities (known also as 2-monotone functions). In fact, not all the uncertainty measures known in literature are decomposable as in Definition 1, nevertheless a measure can be "decomposed" – as in eq. (1) – by an operation satisfying weaker conditions: it means that the properties required to the operation $\oplus$ hold only on suitable subsets, in the sense that, if φ admits an operation $\oplus$ satisfying condition (1), then $\oplus$ is, with respect to the elements of the set

$$\mathcal{K} = \{(\varphi(A), \varphi(B)) : A, B \in \mathcal{E}, A \wedge B = \emptyset\} \subset \varphi(\mathcal{E}) \times \varphi(\mathcal{E}) \,,$$

commutative, associative and admits $\varphi(\emptyset)$ as neutral element, and it is also monotone in some subset of $\mathcal{K}$, depending on the particular choice of φ.

We recall the definition of weakly decomposable measure introduced in [5].

Definition 2. *Let $\mathcal{E}$ be a Boolean algebra of events; a function φ from $\mathcal{E}$ to $[0,1]$ is a* capacity *if $\varphi(\Omega) = 1$, $\varphi(\emptyset) = 0$ and for every $A \subseteq B$ we have $\varphi(A) \leq \varphi(B)$.*

Definition 3. *Let $\mathcal{E}$ be a Boolean algebra of events; a function φ from $\mathcal{E}$ to $[0,1]$ is a weakly $\oplus$-decomposable measure if $\varphi(\emptyset) = 0, \varphi(\Omega) = 1$ and there exists a binary operation $\oplus$ from $\varphi(\mathcal{E}) \times \varphi(\mathcal{E})$ to $\varphi(\mathcal{E})$, whose restriction to the set $\mathcal{K}$ is increasing and is such that condition (1) holds.*

As already noted, the operation $\oplus$ is, with respect to the elements of $\mathcal{K}$, commutative and associative, and by monotonicity requirement, it admits 0 as neutral element. Nevertheless, as proved in [5], $\oplus$ needs not be extendible to a function defined on the whole $\varphi(\mathcal{E}) \times \varphi(\mathcal{E})$ (and so neither in $[0,1]^2$) satisfying the same properties .

In [7], by relaxing the requirement of monotonicity for $\oplus$ on $\mathcal{K}$, a weaker notion (that holds for a large class of uncertainty measures) has been introduced.

Definition 4. *Let $\mathcal{E}$ be a Boolean algebra of events; a function φ from $\mathcal{E}$ to $[0,1]$ is a generalized $\oplus$-decomposable measure if it is a capacity and there exists a binary operation $\oplus$ from $\varphi(\mathcal{E}) \times \varphi(\mathcal{E})$ to $\varphi(\mathcal{E})$, whose restriction to the set $\mathcal{K}$ is such that condition (1) holds.*

The condition on φ to be a capacity implies that $\oplus$ is increasing when restricted to the pairs $\{(\varphi(F), \varphi(E)), (\varphi(F), \varphi(\emptyset)) \, : \, E \wedge F = \emptyset\}$, i.e.

$$\varphi(F) \oplus \varphi(E) = \varphi(E \vee F) \geq \varphi(F) = \varphi(F) \oplus \varphi(\emptyset).$$

Moreover, $\oplus$ admits $\varphi(\emptyset)$ as neutral element in $\mathcal{K}$, while it is not assured that $\varphi(E) \oplus \varphi(F) = \varphi(F)$ even if $\varphi(E) = 0$ (so 0 is not the neutral element).

The class of generalized $\oplus$-decomposable measures includes, for example, belief functions (see [7]), which satisfy the further property that the operation $\oplus$ is strictly increasing with respect to the pairs of events of $\mathcal{K}' \subseteq \mathcal{K}$, with

$$\mathcal{K}' = \{(\varphi(A),\varphi(C)), (\varphi(B), \varphi(C)) : A, B, C \in \mathcal{E}, A \subset B, B \wedge C = \emptyset, \varphi(A) < \varphi(B)\}.$$

As discussed in [10], even convex capacities, i.e. functions such that $\varphi(A \vee B) \geq \varphi(A) + \varphi(B) - \varphi(A \wedge B)$ for any $A, B \in \mathcal{E}$, can be seen as generalized $\oplus$-decomposable measures strictly increasing with respect to the pairs of events of the set $\mathcal{K}'$.

In [10] it has been proved that, inside the class of generalized $\oplus$-decomposable measures strictly increasing with respect to pairs of events in

$$\mathcal{K}_* = \{(\varphi(\emptyset), \varphi(A)), (\varphi(A), \varphi(B)) : A, B \in \mathcal{E},\ A \wedge B = \emptyset,\ \varphi(\emptyset) < \varphi(A)\},$$

there are also 0-monotone functions, i.e. those such that for any $A, B \in \mathcal{E}$, with $A \wedge B = \emptyset$,

$$\varphi(A \vee B) \geq \varphi(A) + \varphi(B).$$

Moreover, it is shown that not all generalized $\oplus$-decomposable measures strictly increasing on $\mathcal{K}'$ [$\mathcal{K}_*$] are 0-monotone; it means that the former classes are strictly larger than the latter.

2.3 Weakly and Generalized Decomposable Conditional Measures

We give now the definition of generalized [weakly] decomposable conditional (uncertainty) measure (introduced in [5] and [7] having in mind that an operation $\odot$ should not be distributive over $\oplus$ on the whole $\varphi(\mathcal{C}) \times \varphi(\mathcal{C})$, but only when we restrict the domain of $\oplus$ to the set $\mathcal{K}$).

Definition 5. *A real function φ defined on $\mathcal{C} = \mathcal{E} \times \mathcal{H}$, with $\mathcal{E}$ a Boolean algebra and $\mathcal{H}$ an additive set, such that $\emptyset \notin \mathcal{H}$, is a* generalized *[weakly] $(\oplus, \odot)$-decomposable conditional measure if*

(C1) $\varphi(E|H) = \varphi(E \wedge H|H)$, for every $E \in \mathcal{E}$ and $H \in \mathcal{H}$,

(C2) for any $H \in \mathcal{H}$ the function $\varphi(\cdot|H)$ is a generalized [weakly] $\oplus$-decomposable measure,

(C3) there exists an operation $\odot : \varphi(\mathcal{C}) \times \varphi(\mathcal{C}) \to \varphi(\mathcal{C})$ whose restriction to the set

$$\Gamma = \{(\varphi(E|H), \varphi(A|E \wedge H)) : A, E \in \mathcal{E},\ H, E \wedge H \in \mathcal{H}\}$$

is increasing, admits 1 as neutral element and is such that, for every $A, E \in \mathcal{E}$ and $E \wedge H, H \in \mathcal{H}$

$$\varphi(E \wedge A|H) = \varphi(E|H) \odot \varphi(A|E \wedge H),$$

(C4) the operation $\odot$ is distributive over $\oplus$ only on the following set

$$\{(\varphi(H|K), (\varphi(E|H \wedge K), \varphi(F|H \wedge K))) : E \wedge F \wedge H \wedge K = \emptyset,\ H \wedge K, K \in \mathcal{H}\}.$$

Notice that the operations $\odot$ is commutative and associative with respect to the elements of Γ.

2.4 Partial Assessments and Coherence

All the above definitions are based on the assumption that the set of events is a Boolean algebra and the set of conditional events is the cartesian product of an algebra and an additive set. Nevertheless, in most situations the base of knowledge is given by an arbitrary set of (conditional) events, and the function φ on them summarizes the state of information. Making inference means enlarging this assessment to new events, maintaining the rules required to the function φ to represent uncertainty (for instance, to be a conditional probability). The notion of coherence formally deals with this problem.

Definition 6. *Given an arbitrary set of conditional events $\mathcal{F}$, a real function φ on $\mathcal{F}$ is a* coherent *generalized [weakly] conditional $(\oplus, \odot)$-decomposable assessment if there exists $\mathcal{C} \supseteq \mathcal{F}$, with $\mathcal{C} = \mathcal{E} \times \mathcal{H}$ (where $\mathcal{E}$ is a Boolean algebra and $\mathcal{H}$ an additive set), and a generalized [weakly] $(\oplus, \odot)$-decomposable conditional measure $\varphi'(\cdot|\cdot)$ on $\mathcal{C}$ extending φ.*

Characterizations of φ for coherent conditional probabilities have been given in [2,4,6], for coherent conditional possibilities in [9,12], for coherent conditional plausibility and belief functions in [3], and, finally, for coherent weakly $(\oplus, \odot)$-decomposable conditional measures in [5].

3 Weak Implication

Given (as measure of reference) a particular coherent conditional measure belonging to one of the classes introduced in Section 2, we adopt (taking into account the considerations made in Sect. 2.4) the following definition of *weak implication*:

Definition 7. *An event A weakly implies an event C under φ (in symbols $A \longmapsto_{\varphi} C$) iff $\varphi(C|A) = 1$.*

We denote by Δ_{φ} the set of given weak implications.

3.1 Weak Implication for Generalized Decomposable Conditional Measures

First of all we notice that the single assessment $\varphi(C|A) = 1$ is coherent (not only for events $A \subseteq C$) for every generalized unconditional measure, except when A and C are incompatible. We remark that we can assign $\varphi(C|A) = 1$ also in the case $\varphi(C|\Omega) = 0$: then the only coherent value for $\varphi(A|\Omega)$ will be 0 (as for conditional probability).

Finally, we recall that for any assessment $\varphi(\mathcal{F})$ which is a coherent generalized [weakly] $(\oplus, \odot)$-decomposable conditional assessment, its enlargement to $\mathcal{C} = \mathcal{E} \times \mathcal{H} \supseteq \mathcal{F}$, is not unique (in general). Nevertheless for some events we can have a unique coherent extension, so giving rise to the important concept of *entailment*.

Definition 8. *If $\varphi(\mathcal{F})$ contains a set of weak implications Δ_φ and for some event $E|H \in \mathcal{F}$ every extension necessarily assumes value equal to 1, then we say that "Δ_φ* entails *$H \longmapsto_\varphi E$"*

Theorem 1. *Let $\mathcal{F}$ be a set of conditional events, $\varphi(\mathcal{F})$ an assessment coherent with a a generalized [weakly] $(\oplus, \odot)$-decomposable conditional measure φ and $\Delta_\varphi \subseteq \varphi(\mathcal{F})$ be the set of the relevant weak implications. Then the following conditions hold:*

(i) *Δ_φ entails $A \longmapsto_\varphi A$ for any $A \neq \emptyset$*

(ii) *$(A = B)\,, (A \longmapsto_\varphi C) \in \Delta_\varphi$ entails $B \longmapsto_\varphi C$*

(iii) *$(A \subseteq B)\,, (C \longmapsto_\varphi A) \in \Delta_\varphi$ entails $C \longmapsto B$*

(iv) *$(A \wedge B \longmapsto_\varphi C)\,, (A \longmapsto_\varphi B) \in \Delta_\varphi$ entails $A \longmapsto_\varphi C$*

Proof. **(i)** amounts to $\varphi(A|A) = 1$ for every possible event A.

(ii) and **(iii)** trivially follow from elementary properties, in particular monotonicity with respect to inclusion of generalized decomposable conditional measures.

(iv) : from $\varphi(C|A \wedge B) = \varphi(B|A) = 1$ it follows that $\varphi(C|A) = 1$, since

$$\varphi(C|A) \geq \varphi(C|A \wedge B) \odot \varphi(B|A) = 1 \odot 1 = 1\,.$$

3.2 Dual Weak Implication

We recall that a function $\varphi^*(\cdot|\cdot)$ is said the dual of a function $\varphi(\cdot|\cdot)$ if for every $E|H \in \mathcal{E} = \mathcal{B} \times \mathcal{H}$ one has:

$$\varphi^*(E|H) = 1 - \varphi(E^c|H),$$

where E^c is the contrary of E.

In general, for a generalized [weakly] $(\oplus, \odot)$-decomposable conditional measure φ the condition $\varphi^*(E|H) = 1$ (which is equivalent to $\varphi(E^c|H) = 0$) does not hold, in other words $\Delta_\varphi \neq \Delta_{\varphi^*}$.

A stronger definition of weak implication is the following

Definition 9. *An event A weakly implies an event C dually under φ (in formulas $A \rightrightarrows_\varphi C$) iff $\varphi(C|A) = \varphi^*(C|A) = 1$.*

We denote by $\Delta_{(\varphi,\varphi^*)}$ the set of given dual weak implications.

Obviously, for $\rightrightarrows_\varphi$ all the properties given in Theorem 1 hold; moreover for $\rightrightarrows_\varphi$ we have:

Theorem 2. *Let $\mathcal{F}$ be a set of conditional events, $\varphi(\mathcal{F})$ an assessment coherent with a weakly $(\oplus, \odot)$-decomposable conditional measure φ and $\Delta_{(\varphi,\varphi^*)} \subseteq \varphi(\mathcal{F})$ be the set of the relevant dual weak implications. Then the following conditions hold:*

(v) *$(A \rightrightarrows_\varphi B)\,, (A \rightrightarrows_\varphi C) \in \Delta_{(\varphi,\varphi^*)}$ entails $A \wedge B \rightrightarrows_\varphi C$;*

(vi) *$(A \rightrightarrows_\varphi B)\,, (B \rightrightarrows_\varphi A)\,, (A \rightrightarrows_\varphi C) \in \Delta_{(\varphi,\varphi^*)}$ entails $B \rightrightarrows_\varphi C$;*

(vii) *$(A \rightrightarrows_\varphi B)\,, (A \rightrightarrows_\varphi C) \in \Delta_{(\varphi,\varphi^*)}$ entails $A \rightrightarrows_\varphi B \wedge C$*

Proof. If $\varphi(B|A) = \varphi(C|A) = 1$ and $\varphi^*(B|A) = \varphi^*(C|A) = 1$, then $\varphi(B^c|A) = \varphi(C^c|A) = 0$ and, since φ is a weakly $(\oplus, \odot)$-decomposable conditional measure, then 0 is the neutral element of $\oplus$, and

$$1 = \varphi(C|A) = \varphi(C \wedge B|A) \oplus \varphi(C \wedge B^c|A) = \varphi(C \wedge B|A) =$$

$$\varphi(C|A \wedge B) \odot \varphi(B|A) = \varphi(C|A \wedge B).$$

Moreover, $\varphi^*(C|A) = 1$ implies $\varphi(C^c \wedge B|A) = 0$, so it follows

$$0 = \varphi(C^c \wedge B|A) = \varphi(C^c|A \wedge B) \odot \varphi(B|A) = \varphi(C^c|A \wedge B)$$

that implies $\varphi^*(C|A \wedge B) = 1 - \varphi(C^c|A \wedge B) = 1$. Then **(v)** holds.

If $\varphi(B|A) = \varphi(A|B) = \varphi(C|A) = 1$ and $\varphi^*(B|A) = \varphi^*(A|B) = \varphi^*(C|A) = 1$, then $\varphi(B^c|A) = \varphi(A^c|B) = \varphi(C^c|A) = 0$ and, analogously to the previous case, since 0 is the neutral element of $\oplus$,

$$1 = \varphi(C|A) = \varphi(C \wedge B|A) \oplus \varphi(C \wedge B^c|A) = (\varphi(C|A \wedge B) \odot \varphi(B|A)) = \varphi(C|A \wedge B)$$

that implies $\varphi(C|B) = 1$, since

$$\varphi(C|B) = \varphi(C \wedge A|B) \oplus \varphi(C \wedge A^c|B) = \varphi(C|A \wedge B) \odot \varphi(A|B) = \varphi(C|A \wedge B).$$

Analogously, $\varphi(C^c|A \wedge B) = \varphi(C^c|A) = 0$ and $\varphi(C^c|B)) = \varphi(C^c|A \wedge B) = 0$, so $\varphi^*(C|B) = 1$. Then **(vi)** holds.

If $\varphi(B|A) = \varphi(C|A) = 1$ and $\varphi^*(B|A) = \varphi^*(C|A) = 1$, then, by duality and monotonicity with respect to inclusion, $\varphi(B \wedge C^c|A) = 0$, so, since 0 is the neutral element of $\oplus$,

$$1 = \varphi(B|A) = \varphi(B \wedge C|A) \oplus \varphi(B \wedge C^c|A) = \varphi(B \wedge C|A).$$

Moreover, again

$$\varphi^*(B \wedge C|A) = 1 - \varphi(B^c \vee C^c|A) = 1 - (\varphi(B^c|A) \oplus \varphi(B \wedge C^c|A)) = 1.$$

Then **(vii)** holds.

The same properties do not hold in general for $\longmapsto_\varphi$, as the following examples show:

Example 1. Consider the following possibility over the algebra generated by the logical independent events A, B, C

$$\Pi(A \wedge B \wedge C) = \Pi(A \wedge B^c \wedge C^c) = \frac{1}{4}\,,$$

$$\Pi(A \wedge B \wedge C^c) = \Pi(A \wedge B^c \wedge C) = \Pi(A^c \wedge B \wedge C^c) = \Pi(A^c \wedge B^c \wedge C) = \frac{1}{2}\,,$$

$$\Pi(A^c \wedge B \wedge C) = \Pi(A^c \wedge B^c \wedge C^c) = 1\,,$$

hence

$$\Pi(A) = \Pi(A \wedge B) = \Pi(A \wedge C) = \frac{1}{2} .$$

Taking as triangular norm T the ordinary product we obtain a T-conditional possibility $\Pi(\cdot|\cdot)$ such that $\Pi(B|A) = \Pi(C|A) = 1$, while $\Pi(C|A \wedge B) = \frac{1}{2}$.

This shows that $A \longmapsto_\Pi B$ and $A \longmapsto_\Pi C$, but $A \wedge B \longmapsto_\Pi C$ fails.

Note that the same conclusion follows by taking as triangular norm the minimum (which is not strict).

Example 2. Consider the following possibility over the algebra generated by the logical independent events A, B, C

$$\Pi(A \wedge B \wedge C) = \Pi(A^c \wedge B \wedge C) = \frac{1}{4} ,$$

$$\Pi(A \wedge B \wedge C^c) = \Pi(A \wedge B^c \wedge C) = \frac{1}{2} ,$$

$$\Pi(A \wedge B^c \wedge C^c) = \Pi(A^c \wedge B \wedge C^c) = \Pi(A^c \wedge B^c \wedge C) = \frac{1}{8} ,$$

$$\Pi(A^c \wedge B^c \wedge C^c) = 1 ,$$

hence $\Pi(A) = \Pi(A \wedge B) = \Pi(A \wedge C) = \Pi(B) = \frac{1}{2}$, and $\Pi(B \wedge C) = \frac{1}{4}$.

Taking as triangular norm T the ordinary product we obtain a T-conditional possibility $\Pi(\cdot|\cdot)$ such that $\Pi(B|A) = \Pi(C|A) = \Pi(A|B) = 1$, but $\Pi(C|B) = \frac{1}{2}$.

Then $A \longmapsto_\Pi B$ and $B \longmapsto_\Pi A$, but $B \longmapsto_\Pi C$ fails.

Example 3. Consider the following possibility over the algebra generated by the logical independent events A, B, C

$$\Pi(A \wedge B \wedge C) = \frac{1}{2} , \ \Pi(A^c \wedge B^* \wedge C^*) = \frac{1}{8} ,$$

where B^* – and analogously for C^* – stands for B or B^c, and

$$\Pi(A \wedge B \wedge C^c) = \Pi(A \wedge B^c \wedge C) = 1 , \ \Pi(A \wedge B^c \wedge C^c) = \frac{1}{8} ,$$

hence, taking for $\odot$ again the ordinary product, $\Pi(B|A) = \Pi(C|A) = 1$, while $\Pi(B \wedge C|A) = \frac{1}{2}$.

Then $A \longmapsto_\Pi B$ and $A \longmapsto_\Pi C$, but $A \longmapsto_\Pi B \wedge C$ fails.

Moreover, the properties **(v), (vi), (vii)** do not necessarily hold for $\rightrightarrows_\varphi$ when φ is a generalized decomposable measure, as the following example shows:

Example 4. Consider the following convex capacity over the algebra $\mathcal{E}$ generated by the logical independent events A, B, C:

$$\varphi(A \wedge B \wedge C) = 0.2 , \ \varphi(A \wedge B \wedge C^c) = \varphi(A \wedge B^c \wedge C) = \varphi(A \wedge B^c \wedge C^c) = 0 ,$$

$$\varphi((A \wedge B \wedge C^c) \vee (A \wedge B^c \wedge C)) = 0 , \ \varphi((A \wedge B \wedge C) \vee (A \wedge B^c \wedge C^c)) = 0.2$$

$$\varphi((A \wedge B \wedge C^c) \vee (A \wedge B^c \wedge C)) = \varphi(A \wedge B^c) = 0\,,$$
$$\varphi(A \wedge B) = \varphi(A \wedge C) = 1\,,$$

so for any $K \in \mathcal{E}$ including $(A \wedge B)$ or $(A \wedge C)$ we have $\varphi(K) = 1$ and $\varphi(K^c) = 0$. By considering any operation $\odot$ having 1 as neutral elements and satisfying conditions $(C3)$ and $(C4)$ of Definition 5, one gets $\varphi(B|A) = \varphi(C|A) = \varphi(A|B) = 1$ and $\varphi(B^c|A) = \varphi(C^c|A) = \varphi(A^c|B) = 0$ (so $\varphi^*(B|A) = \varphi^*(C|A) = \varphi^*(A|B) = 1$), while $\varphi(C|A \wedge B) = 0.2 = \varphi(C|B) = \varphi(B \wedge C|A)$.

Then, the statements **(v), (vi)** and **(vii)** do not hold.

Given a conditional measures φ, if for any $H \in \mathcal{H}$ the measure $\varphi(\cdot|H)$ is 0–monotone, i.e. $\varphi(E|H) + \varphi(E^c|H) \leq \varphi(H|H) = 1$, it follows (since $\varphi(E|H) = 1$ implies $\varphi(E^c|H) = 0$) $\varphi^*(E|H) = 1$, and then Δ_φ coincides with $\Delta_{(\varphi,\varphi^*)}$.

Thus, from the above results it comes out that for a weakly $(\oplus, \odot)$-decomposable conditional measure φ on $\mathcal{C} = \mathcal{E} \times \mathcal{H}$ such that, for any $H \in \mathcal{H}$, the measure $\varphi(\cdot|H)$ is 0–monotone, Δ_φ satisfies all the properties **(i)** –**(vii)**.

3.3 Weak Implication for Probabilities

Since a conditional probability is a particular weakly decomposable conditional measure, and for any given conditioning event it is 0–monotone, it follows that if Δ_P is a set of weak implications induced by a conditional probability P (i.e. $H \longmapsto_P E$ means $P(E|H) = 1$) we have that properties **(i)** –**(vii)** hold.

Moreover, it satisfies a further property:

Theorem 3. *Given a set Δ_P of weak implication induced by a coherent conditional probability, we have*

(viii) $(A \longmapsto_P C)\,, (B \longmapsto_P C) \in \Delta_P$ *entails* $A \vee B \longmapsto_P C$.

Proof. Since

$$P(C|A \vee B) =$$
$$= P(C|A)P(A|A \vee B) + P(C|B)P(B|A \vee B) - P(C|A \wedge B)P(A \wedge B|A \vee B) =$$
$$= P(A|A \vee B) + P(B|A \vee B) - P(C|A \wedge B)P(A \wedge B|A \vee B) \geq 1\,,$$

we get $P(C|A \vee B) = 1$.

Remark. *Properties* **(i)**–**(viii)** *correspond to those that, in default logic (see, e.g., [13]), are called, respectively,* Reflexivity, Left Logical Equivalence, Right Weakening, Cut, Cautious Monotonicity, Equivalence, And, Or.

Many authors (cfr., e.g., [13]) claim (and we agree) that the following properties should be regarded as "unpleasant" (they in fact do not necessarily hold in our framework: see [8]):

(Monotonicity)
$(A \subseteq B)\,, (B \longmapsto_P C) \in \Delta_P$ entails $A \longmapsto_P C$

(Transitivity)
$(A \longmapsto_P B)\,, (B \longmapsto_P C) \in \Delta_P$ entails $A \longmapsto_P C$

(Contraposition)

$(A \longmapsto_P B) \in \Delta_P$ entails $B^c \longmapsto_P A^c$

So they should be replaced by others, that we express below in our own notation and interpretation: we show that these properties hold in our framework. Since a widespread consensus among their "right" formulation is lacking, we will denote them as cs–(Negation Rationality), cs–(Disjunctive Rationality), cs–(Rational Monotonicity), where "cs" stands for "in a coherent setting".

Notice that, given a weak implication $H \longmapsto_P E$, to say $(H \longmapsto_P E) \notin \Delta_P$ means that the conditional event $E|H$ belongs to the set $\mathcal{C} \setminus \Delta_P$.

cs–(Negation Rationality)

If $(A \wedge C \longmapsto_P B)$, $(A \wedge C^c \longmapsto_P B) \notin \Delta_P$ then Δ_P does not entail $(A \longmapsto_P B)$.

Proof. Since $(A \wedge C \longmapsto_P B)$ and $(A \wedge C^c \longmapsto_P B)$ do not belong to Δ_P, that is $P(B|A \wedge C) < 1$ and $P(B|A \wedge C^c) < 1$, then

$$P(B|A) = P(B|A \wedge C)P(C|A) + P(B|A \wedge C^c)P(C^c|A) < 1\,.$$

cs–(Disjunctive Rationality)

If $(A \longmapsto_P C)$, $(B \longmapsto_P C) \notin \Delta_P$ then Δ_P does not entail $(A \vee B \longmapsto_P C)$.

Proof. Starting from the equalities

$$P(C|A \vee B) = P(C|A)P(A|A \vee B) + P(C|A^c \wedge B)P(A^c \wedge B|A \vee B) =$$

$$= P(C|B)P(B|A \vee B) + P(C|A \wedge B^c)P(A \wedge B^c|A \vee B),$$

since $P(C|A) < 1$ and $P(C|B) < 1$, assuming $P(C|A \vee B) = 1$ would imply (by the first equality) $P(A|A \vee B) = 0$ and (by the second one) $P(B|A \vee B) = 0$ (contradiction).

cs–(Rational Monotonicity)

If $(A \wedge B \longmapsto_P C)$, $(A \longmapsto_P B^c) \notin \Delta_P$ then Δ_P does not entail $(A \longmapsto_P C)$.

Proof. If it were $P(C|A) = 1$, i.e.

$$1 = P(C|A \wedge B)P(B|A) + P(C|A \wedge B^c)P(B^c|A)\,,$$

we would get either $P(C|A \wedge B) = P(C|A \wedge B^c) = 1$ or one of the following

$$P(C|A \wedge B) = P(B|A) = 1\,,\ P(C|A \wedge B^c) = P(B^c|A) = 1$$

(contradiction).

4 Conclusions

The theory of default reasoning has been embedded in a general theory concerning weak implication (an event H weakly implies an event E), expressed through the relation $\varphi(E|H) = 1$ and considering various coherent conditional uncertainty measures φ with different levels of generality. One of the main result is that in our framework default logic can be seen as a particular case of weak implication when we take as φ a coherent conditional probability.

References

1. Bouchon-Meunier, B., Coletti, G., Marsala, C.: Independence and Possibilistic Conditioning. Annals of Mathematics and Artificial Intelligence 35, 107–124 (2002)
2. Coletti, G.: Coherent numerical and ordinal probabilistic assessments. IEEE Transactions on Systems, Man, and Cybernetics 24, 1747–1754 (1994)
3. Coletti, G., Mastroleo, M.: Conditional belief functions: a comparison among different definitions. In: Proc. of WUPES 2006, Mikulov (Czech Rep.), pp. 24–35 (2006)
4. Coletti, G., Scozzafava, R.: Characterization of coherent conditional probabilities as a tool for their assessment and extension. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 4, 103–127 (1996)
5. Coletti, G., Scozzafava, R.: From conditional events to conditional measures: a new axiomatic approach. Annals of Mathematics and Artificial Intelligence 32, 373–392 (2001)
6. Coletti, G., Scozzafava, R.: Probabilistic logic in a coherent setting. In: Trends in logic, vol. 15, Kluwer, Dordrecht (2002)
7. Coletti, G., Scozzafava, R.: Toward a general theory of conditional beliefs. International Journal of Intelligent Systems 21, 229–259 (2006)
8. Coletti, G., Scozzafava, R., Vantaggi, B.: Coherent conditional probability as a tool for default reasoning. In: Bouchon-Meunier, B., Foulloy, L., Yager, R.R. (eds.) Intelligent Systems for Information Processing: from Representation to Applications, pp. 191–202. Elsevier, Amsterdam (2003)
9. Coletti, G., Vantaggi, B.: Possibility theory: conditional independence. Fuzzy Sets and Systems 157, 1491–1513 (2006)
10. Coletti, G., Vantaggi, B.: Representability of ordinal relations on a set of conditional events. Theory and Decision 60, 137–174 (2006)
11. de Finetti, B.: Sull'impostazione assiomatica del calcolo delle probabilità. Annali Univ. Trieste 19, 3–55 (1949) (Engl. transl.: Probability, Induction, Statistics, ch. 5, Wiley, London (1972))
12. Ferracuti, L., Vantaggi, B.: Independence and conditional possibilities for strictly monotone triangular norms. International Journal of Intelligent Systems 21(3), 299–323 (2006)
13. Lehmann, D., Magidor, M.: What does a conditional knowledge base entail? Artificial Intelligence 55, 1–60 (1992)
14. Paneni, T., Scozzafava, R.: Multi-valued conditional events avoid Lewis' triviality result. In: Nielsen, T.D., Zhang, N.L. (eds.) ECSQARU 2003. LNCS (LNAI), vol. 2711, pp. 432–439. Springer, Heidelberg (2003)
15. Reiter, R.: A Logic for Default Reasoning. Artificial Intelligence 13, 81–132 (1980)
16. Russel, S.J., Norvig, P.: Artificial Intelligence, a Modern Approach. Prentice-Hall, New Jersey (1995)

Language Invariance and Spectrum Exchangeability in Inductive Logic

Jürgen Landes*, Jeff Paris, and Alena Vencovská**

School of Mathematics
University of Manchester
Manchester M13 9PL, UK
juergen_landes@yahoo.de, jeff.paris@manchester.ac.uk,
alena.vencovska@manchester.ac.uk

Abstract. A sufficient condition, in terms of a de Finetti style representation, is given for a probability function in Inductive Logic (with relations of all arities) satisfying Spectrum Exchangeability to additionally satisfy Language Invariance. This condition is shown to also be necessary in the case of homogeneous probability functions. In contrast it is proved that (purely) t-heterogeneous probability functions can never be members of a language invariant family satisfying Spectrum Exchangeability.

Keywords: Uncertain reasoning, inductive logic, probability logic, Spectrum Exchangeability, Language Invariance.

1 Introduction

In common with recent developments in Inductive Logic, see for example [15] (and [1], [2], [3] for the classical approach), we shall work within a first order predicate language L with finitely many relation symbols, countably many constants $a_1, a_2, a_3, \ldots$ and no function symbols. The intention here is that these constants a_i exhaust the universe. Let $SL, QFSL$ respectively denote the sentences and quantifier free sentences of L.

We say that a function $w : SL \to [0, 1]$ is a *probability function* on L if it satisfies that for all $\theta, \phi, \exists x\, \psi(x) \in SL$:

(P1) If $\vDash \theta$ then $w(\theta) = 1$
(P2) If $\vDash \neg(\theta \wedge \phi)$ then $w(\theta \vee \phi) = w(\theta) + w(\phi)$
(P3) $w(\exists x\, \psi(x)) = \lim_{m\to\infty} w(\bigvee_{i=1}^{m} \psi(a_i))$

(P1) and (P2) are the standard axioms for a probability function. (P3) is the Gaifman axiom (see [4]) expressing the fact that the a_i exhaust the universe and that in consequence $\exists x\, \psi(x)$ should be equated with the infinite disjunction $\bigvee_{i=1}^{\infty} \psi(a_i)$.

* Supported by a MATHLOGAPS Research Studentship.
** Supported by a UK Engineering and Physical Sciences Research Council (EPSRC) Research Assistantship.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 151–160, 2007.

Throughout w, possibly with various annotations, will denote a probability function on L and, for the purposes of motivation, we shall be thinking of probabilities in the sense of de Finetti as subjective degrees of belief.

By a theorem of Gaifman (see [4], where in fact the axioms (P1-3) were first formulated) any probability function defined on $QFSL$ (i.e. satisfying (P1) and (P2) for $\theta, \phi \in QFSL$) extends uniquely to a probability function on L. In this sense then we can largely limit our considerations to probability functions defined just on $QFSL$. Indeed, by the Disjunctive Normal Form Theorem it then follows that w is determined simply by its values on the *state descriptions*, that is sentences of the form

$$\bigwedge_{s=1}^{m} \bigwedge_{c_1,c_2,\ldots,c_{r_s} \in \{b_1,b_2,\ldots,b_n\}} \pm P_s(c_1, c_2, ..., c_{r_s})$$

where the b_i are distinct constants from L (i.e. choices of a_j) and $P_1, P_2, \ldots, P_m$ are the relations of L with arities $r_1, r_2, \ldots, r_m$ respectively.

In Inductive Logic we are basically interested in the choice of probability functions w on L when these are intended to represent the beliefs, i.e subjective probabilities, assigned by a *rational* or *logical* agent in the absence of any prior knowledge. The key restraint here is that this assignment should be *rational* or *logical* and it is customary to identify this with the requirement that w satisfies certain rational or common sense principles.[1]

A number of such principles have been suggested, see for example [1], [2], [3], [5], [8], [11], [13], [14], [15], [17], the most basic of which asserts that w should not treat the constants a_i differently:

The Constant Exchangeability Principle (Ex)
For $\theta, \theta' \in QFSL$, if θ' is obtained from θ by replacing, respectively, the (distinct) constant symbols $b_1, b_2, \ldots, b_k$ from L occurring in θ by the (distinct) constant symbols $c_1, c_2, \ldots, c_k$ from L then $w(\theta) = w(\theta')$.

We remark that for a purely unary language L this principle corresponds to the property of exchangeability in the sense of de Finetti. For a purely n-ary language (with a fixed $n > 1$) the corresponding property is that of joint exchangeability studied by Kallenberg et al, see [6], [7]. We shall henceforth assume that all probability functions mentioned satisfy Ex. Notice that in this case the value of w on the state description as above does not depend on the choice of $b_1, b_2 \ldots, b_n$ from amongst the a_j and so we may without loss take these to be $a_1, a_2, \ldots, a_n$.

There are two other such 'common sense' principles which we shall be concerned with in this paper. In order to explain the first of these we need to

[1] The relevance of the results in this paper to *practical* uncertain reasoning is largely of a cautionary nature. We show how certain, arguably rational, general requirements on the assignment of beliefs as probabilities in fact impose quite stringent practical limits on the choices permitted. Thus the non-observance of this practical straightjacket amounts to flouting rationality, as we have interpreted it.

introduce some notation. Given a state description $\Theta(b_1, b_2, \ldots, b_n)$ where the b_i are distinct constants from L we say that b_i, b_j are *indistinguishable* mod Θ, written $b_i \sim_\Theta b_j$, if for any relation $P(x_1, x_2, \ldots, x_r)$ of L and any $t_1, \ldots, t_r \in \{1, \ldots, n\}$, the sentence $P(b_{t_1}, b_{t_2}, \ldots, b_{t_r})$ appears positively as a conjunct in $\Theta(b_1, b_2, \ldots, b_n)$ if and only if $P(b_{s_1}, b_{s_2}, \ldots, b_{s_r})$ also appears positively as a conjunct in $\Theta(b_1, b_2, \ldots, b_n)$ where $\langle b_{s_1}, b_{s_2}, \ldots, b_{s_r}\rangle$ is the result of replacing any number of occurrences of b_i in $\langle b_{t_1}, b_{t_2}, \ldots, b_{t_r}\rangle$ by b_j or vice-versa. Clearly $\sim_\Theta$ is an equivalence relation.

Define the *spectrum* of Θ, denoted $\mathcal{S}(\Theta)$, to be the multiset[2] of sizes of the (non-empty) equivalence classes with respect to $\sim_\Theta$.

The Spectrum Exchangeability Principle (Sx)
If $\Theta(b_1, b_2, \ldots, b_n), \Phi(c_1, c_2, \ldots, c_n)$ are state description and $\mathcal{S}(\Theta) = \mathcal{S}(\Phi)$ then $w(\Theta) = w(\Phi)$.

Clearly expressed in this form Sx implies Ex. In the early accounts of Inductive Logic, for example [1], [2], [3], [8], the language L was taken to be purely unary, that is the predicates of the language are just $P_1(x), P_2(x), \ldots, P_s(x)$ (but see [9], [10]). In this case state descriptions have the simple form

$$\bigwedge_{i=1}^{n} \alpha_{h_i}(b_i)$$

where the $\alpha_h(x)$, $h = 1, 2, \ldots, 2^s$ are the *atoms* of L, that is formulae of the form

$$\pm P_1(x) \wedge \pm P_2(x) \wedge \ldots \wedge \pm P_s(x),$$

and Sx reduces to *Atom Exchangeability*, Ax, asserting that

$$w\left(\bigwedge_{i=1}^{n} \alpha_{h_i}(b_i)\right)$$

depends only on the multiset of $|\{i \,|\, h_i = j\}|$ for $j = 1, 2, \ldots, 2^s$.

It appears that the principle Ax was acceptable to Johnson and Carnap and the earlier investigators since it follows from Johnson's Sufficientness Principle[3] which they advocated.

The second principle which we shall be concerned with here is that of Language Invariance. The motivation behind this principle is that whilst we may at any one time be interested is some particular finite language L a rational choice of beliefs for that language should be capable of extension to a larger language. After all there is clearly no reason to suppose that there are only finitely many relations in existence and that L has already included all of them.

[2] Equivalently, we can define the spectrum as the vector of sizes of the (non-empty) equivalence classes in non-decreasing order.

[3] See for example [17] or [18] for a formulation of this Principle in the notation of this paper.

Language Invariance

The probability function w on L satisfies *Language Invariance*[4] if there exist a class of probability functions $w_{\mathcal{L}}$ for each finite predicate language $\mathcal{L}$ (as usual with constants a_i and no function symbols) such that whenever $\mathcal{L}'$ is a sublanguage of $\mathcal{L}$ then w restricted to $S\mathcal{L}'$ equals $w_{\mathcal{L}'}$ ($w_{\mathcal{L}} \restriction S\mathcal{L} = w_{\mathcal{L}'}$) and $w_L = w$.

In this case we shall describe the $w_{\mathcal{L}}$ as a *language invariant family* containing w.

In the next section we shall derive a sufficient condition for a probability function satisfying Spectrum Exchangeability to also satisfy Language Invariance.

2 A Sufficiency Condition for Language Invariance

Before stating and proving the main result of this paper we need to introduce a particular family of probability functions $u_L^{\overline{p}}$.

Let

$$\mathbb{B} = \{\langle x_0, x_1, x_2, ...\rangle \,|\, x_1 \geq x_2 \geq ... \geq 0,\, x_0 \geq 0,\, \sum_{i=0}^{\infty} x_i = 1\}$$

and endow $\mathbb{B}$ with the standard weak product topology inherited from $[0,1]^\infty$. Let

$$\overline{p} = \langle p_0, p_1, p_2, \ldots\rangle \in \mathbb{B}.$$

For a state description $\Theta(b_1, b_2, \ldots, b_q)$ (from language L) and 'colours'

$$\boldsymbol{c} = \langle c_1, c_2, \ldots, c_q\rangle \in \{0, 1, 2, \ldots\}^q$$

(where 0 stands for the special colour black) we define $j^{\overline{p}}(\Theta(b_1, b_2, \ldots, b_q), \boldsymbol{c})$ inductively as follows:

Set $j^{\overline{p}}(\top, \emptyset) = 1$. Suppose that at stage q we have defined the probability $j^{\overline{p}}(\Theta(b_1, b_2, \ldots, b_q), \boldsymbol{c})$. Pick colour c_{q+1} from $0, 1, 2, \ldots$ according to the probabilities $p_0, p_1, p_2, \ldots$ and let

$$\boldsymbol{c}^+ = \langle c_1, \ldots, , c_q, c_{q+1}\rangle.$$

If c_{q+1} is the same as an earlier colour, c_j say, with $c_j \neq 0$ extend $\Theta(b_1, b_2, \ldots, b_q)$ to the unique state description $\Theta^+(b_1, b_2, \ldots, b_q, b_{q+1})$ for which $b_j \sim_{\Theta^+} b_{q+1}$. On the other hand if c_{q+1} is 0 or a previously unchosen colour then randomly choose $\Theta^+(b_1, b_2, \ldots, b_q, b_{q+1})$ extending $\Theta(b_1, b_2, \ldots, b_q)$ such that $\sim_\Theta$ and $\sim_{\Theta^+}$ agree on $\{b_1, b_2, \ldots, b_q\}$[2]. Finally let $j^{\overline{p}}(\Theta^+, \boldsymbol{c}^+)$ be $j^{\overline{p}}(\Theta, \boldsymbol{c})$ times the probability as described of then going from $\Theta, \boldsymbol{c}$ to $\Theta^+, \boldsymbol{c}^+$.

Having defined these $j^{\overline{p}}(\Theta, \boldsymbol{c})$ now set

$$u_L^{\overline{p}}(\Theta) = \sum_{\boldsymbol{c}} j^{\overline{p}}(\Theta, \boldsymbol{c}).$$

[4] This differs from the earlier definition of Language Invariance given in [5] and [17] which was restricted to purely unary languages $\mathcal{L}$, $\mathcal{L}'$.

By a straightforward generalization of the result in [15] (where just two colours were considered) $u_L^{\overline{p}}$ satisfies Sx (and hence also Ex).

Theorem 1. *Let the probability function w on L satisfy Sx. Then for w to be a member of a language invariant family all satisfying Sx it is sufficient that there is a measure μ on the Borel subsets of $\mathbb{B}$ such that for $\theta \in SL$,*

$$w(\theta) = \int_{\mathbb{B}} u_L^{\overline{p}}(\theta) d\mu. \tag{1}$$

Furthermore in this case if L contains at least one non-unary relation then this language invariant family containing w is unique.

We call μ as in this theorem the *de Finetti prior* of w.

Proof. Suppose that (1) holds. Let $\mathcal{L}$ extend L and for $\phi \in S\mathcal{L}$ set

$$w_{\mathcal{L}}(\phi) = \int_{\mathbb{B}} u_{\mathcal{L}}^{\overline{p}}(\phi) d\mu, \tag{2}$$

in other words $w_{\mathcal{L}}$ has the same de Finetti prior as w, but the language has changed. Since the $u_{\mathcal{L}}^{\overline{p}}$ satisfy Sx, so do the $w_{\mathcal{L}}$. We claim that $w_{\mathcal{L}} \restriction SL = w$. To show this it is enough to show that for a state description $\Theta(\boldsymbol{a})$ from language L, $w_{\mathcal{L}}(\Theta(\boldsymbol{a})) = w(\Theta(\boldsymbol{a}))$, and for this it is enough to show that

$$u_{\mathcal{L}}^{\overline{p}}(\Theta(\boldsymbol{a})) = u_L^{\overline{p}}(\Theta(\boldsymbol{a})). \tag{3}$$

Let $\Phi(\boldsymbol{a})$ be a state description for $\mathcal{L}$ extending $\Theta(\boldsymbol{a})$ (and with the same constants $\boldsymbol{a} = \langle a_1, a_2, \ldots, a_q \rangle$) and consider a summand $j_{\mathcal{L}}^{\overline{p}}(\Phi(\boldsymbol{a}), \boldsymbol{c})$ which yields $u_{\mathcal{L}}^{\overline{p}}(\Theta(\boldsymbol{a}))$ via

$$u_{\mathcal{L}}^{\overline{p}}(\Phi(\boldsymbol{a})) = \sum_{\boldsymbol{c}} j_{\mathcal{L}}^{\overline{p}}(\Phi(\boldsymbol{a}), \boldsymbol{c}).$$

This summand is formed by q choices of colours $c_1, c_2, \ldots, c_q$ and an increasing sequences of choices of state descriptions

$$\Phi_1(a_1), \Phi_2(a_1, a_2), \Phi_3(a_1, a_2, a_3), \ldots, \Phi_q(a_1, a_2, \ldots, a_q) = \Phi(\boldsymbol{a}).$$

Let $\Theta_k(a_1, a_2, \ldots, a_k)$ be the state description of L which $\Phi_k(a_1, a_2, \ldots, a_k)$ extends. Then

$$\Theta_q(a_1, a_2, \ldots, a_q) = \Theta(a_1, a_2, \ldots, a_q)$$

and for this same choice of colours $\boldsymbol{c}$ and Θ_k $j_L^{\overline{p}}(\Theta(\boldsymbol{a}), \boldsymbol{c})$ is a contributing summand to $u_L^{\overline{p}}(\Theta(\boldsymbol{a}))$. Furthermore the only difference between these two contributions is that at each choice of the kth state description $j_{\mathcal{L}}^{\overline{p}}(\Phi(\boldsymbol{a}), \boldsymbol{c})$ receives as a multiplicative factor one over the number of possible state descriptions in $\mathcal{L}$ extending $\Phi_{k-1}(a_1, a_2, \ldots, a_{k-1})$ whereas $j_L^{\overline{p}}(\Theta(\boldsymbol{a}), \boldsymbol{c})$ receives as a multiplicative factor one over the number of possible state descriptions in L extending $\Theta_{k-1}(a_1, a_2, \ldots, a_{k-1})$. Note that this depends on c_k, a genuine choice being

available only when c_k is either zero or not equal to any previous c_j. However, since otherwise in each case these factors depend only on k, $\boldsymbol{c}$ and on the relations in L and $\mathcal{L}$ and not on the particular state descriptions $\Theta_{k-1}(a_1, \ldots, a_{k-1})$, $\Phi_{k-1}(a_1, \ldots, a_{k-1})$, overall

$$j_L^{\overline{p}}(\Theta(\boldsymbol{a}), \boldsymbol{c}) = M j_{\mathcal{L}}^{\overline{p}}(\Phi(\boldsymbol{a}), \boldsymbol{c})$$

where M is the number of possible choices (according to $\boldsymbol{c}$) of state descriptions of $\boldsymbol{a}$ in $\mathcal{L}$ extending $\Theta(\boldsymbol{a})$. But this means that

$$j_L^{\overline{p}}(\Theta(\boldsymbol{a}), \boldsymbol{c}) = \sum_{\Psi(\boldsymbol{a})} j_{\mathcal{L}}^{\overline{p}}(\Psi(\boldsymbol{a}), \boldsymbol{c})$$

where $\Psi(\boldsymbol{a})$ runs over the M many state descriptions of $\boldsymbol{a}$ in $\mathcal{L}$ admitted by $\boldsymbol{c}$ and extending $\Theta(\boldsymbol{a})$. Since

$$u_{\mathcal{L}}^{\overline{p}}(\Theta(\boldsymbol{a})) = \sum_{\Psi(\boldsymbol{a})} \sum_{\boldsymbol{c}} j_{\mathcal{L}}^{\overline{p}}(\Psi(\boldsymbol{a}), \boldsymbol{c})$$

where the sum is over all extensions $\Psi(\boldsymbol{a})$ in $\mathcal{L}$ of $\Theta(\boldsymbol{a})$, rearranging the summation on the right hand side yields

$$u_{\mathcal{L}}^{\overline{p}}(\Theta(\boldsymbol{a})) = u_L^{\overline{p}}(\Theta(\boldsymbol{a})),$$

as required.

Of course the required 'full' language invariant family for w can now be obtained by restricting/marginalizing these $w_{\mathcal{L}}$.

To show uniqueness suppose that L has some non-unary relation symbol and that there are two different language invariant families containing w, say w', w'' are the members of these families defined on $\mathcal{L} \supset L$ and they differ on some state description, $\Psi(a_1, a_2, \ldots, a_n)$ say.

We first define a well founded ordering on state descriptions $\Theta(a_1, a_2, \ldots, a_n)$ of $\mathcal{L}$ or L, for fixed n, by setting

$$\Theta(\boldsymbol{a}) \trianglelefteq \Phi(\boldsymbol{a}) \iff \sim_\Theta \text{ is a refinement of } \sim_\Phi .$$

We now show

$$w'(\Theta(\boldsymbol{a})) = w''(\Theta(\boldsymbol{a})) \tag{4}$$

by induction on this ordering. The least point in this ordering is when the equivalence classes of $\sim_\Theta$ are all singletons. In this case let $\Phi_L(a_1, a_2, \ldots, a_n)$ be a state description of L having this minimal spectrum. (This is where we need L to contain a non-unary relation symbol, to ensure that such a state description exists.) Then $\sim_{\Phi(\boldsymbol{a})}$ must again be this minimal spectrum for any state description $\Phi(a_1, a_2, \ldots, a_n)$ of $\mathcal{L}$ extending $\Phi_L(a_1, a_2, \ldots, a_n)$ and w' must take the same value on these by Sx. Hence, since

$$w(\Phi_L(a_1, a_2, \ldots, a_n)) = w'(\Phi_L(a_1, a_2, \ldots, a_n)) = \sum_{\Phi(\boldsymbol{a})} w'(\Phi(\boldsymbol{a})),$$

where the summation is over state descriptions $\Phi(\boldsymbol{a})$ extending $\Phi_L(\boldsymbol{a})$, we see that if M is the number of such $\Phi(\boldsymbol{a})$ then for any one of them

$$w'(\Phi(a_1, a_2, \ldots, a_n)) = M^{-1} w(\Phi_L(a_1, a_2, \ldots, a_n)).$$

Since this reasoning also applies to w'' (4) holds in this base case.

Now suppose that (4) holds for all $\Phi(\boldsymbol{a}) \lhd \Theta(\boldsymbol{a})$. Let $\Theta_L(\boldsymbol{a})$ be a state description of L having the same spectrum as $\Theta(\boldsymbol{a})$. Then again,

$$w(\Theta_L(\boldsymbol{a})) = \sum_{\Phi(\boldsymbol{a})} w'(\Phi(\boldsymbol{a})) \tag{5}$$

where the $\Phi(\boldsymbol{a})$ range over state descriptions in $\mathcal{L}$ extending $\Theta_L(\boldsymbol{a})$. Now all of these $\Phi(\boldsymbol{a})$ are less or equal to $\Theta(\boldsymbol{a})$ in the ordering $\trianglelefteq$, *and* state descriptions with the same spectrum as $\Theta(\boldsymbol{a})$ do itself appear on the right hand side of this expression a non-zero number of times. Furthermore the identity (5) also holds with the probability function w'' in place of w', and by the inductive hypothesis these terms are the same except possibly for the $w''(\Phi(\boldsymbol{a}))$ when $\Phi(\boldsymbol{a})$ has the same spectrum as $\Theta(\boldsymbol{a})$. But then of course they must also be the same in this case, as required to prove (4) and the theorem.

3 An Application

For this section assume that our default language L has at least one non-unary relation. We first recall a classification[5] of probability functions w on L satisfying Sx.

We say that w is *homogeneous* if for all k

$$\lim_{r\to\infty} \sum_{|\mathcal{S}(\Theta(a_1,a_2,\ldots,a_r))|=k} w(\Theta(a_1, a_2, \ldots, a_r)) = 0$$

where the $\Theta(a_1, a_2, \ldots, a_r)$ range over the possible state descriptions of $a_1, a_2, \ldots, a_r$ in L. In other words the probability that all the a_i will fall in some fixed finite number of equivalence classes with respect to indistinguishability is zero.

We say that w is *t-heterogeneous* if

$$\lim_{r\to\infty} \sum_{|\mathcal{S}(\Theta(a_1,a_2,\ldots,a_r))|=t} w(\Theta(a_1, a_2, \ldots, a_r)) = 1.$$

In other words the probability that all the a_i will fall in some t (non-empty) equivalence classes with respect to indistinguishability is 1.

The following theorem appears in [14], [15] for the case of a purely binary language and will appear in [11] for general not purely unary languages.

[5] Given in [14], [15] for binary languages and more generally in the forthcoming [11].

Theorem 2. *Let w satisfy Sx. Then there are probability functions $w^{[t]}$ satisfying Sx and constants $\eta_t \geq 0$ for $0 \leq t < \infty$ such that*

$$w = \sum_{i=0}^{\infty} \eta_i w^{[i]}, \qquad \sum_{i=0}^{\infty} \eta_i = 1,$$

$w^{[t]}$ is t-heterogeneous for $t > 0$ and $w^{[0]}$ is homogeneous. Furthermore the η_i are unique and so are the $w^{[i]}$ when $\eta_i \neq 0$.

The following result[6] will appear in the forthcoming paper [12].

Theorem 3. *Let w be a homogeneous probability function on L (not purely unary) satisfying Sx. Then there is a measure μ on the Borel subsets of $\mathbb{B}$ such that for $\theta \in SL$,*

$$w(\theta) = \int_{\mathbb{B}} u_L^{\overline{p}}(\theta) d\mu.$$

Using this result we have the following corollary to Theorem 1:

Corollary 1. *Let w be a homogeneous probability function on L (not purely unary) satisfying Sx. Then w satisfies Language Invariance.*

This is, to our way of thinking, a rather surprising result. In fact, considering also the uniqueness part of Theorem 1, it means that just knowing a homogeneous w on a sublanguage consisting of a single non-unary relation is, provided we require Sx to be preserved, enough to determine it on all extensions of that language.

In contrast to Corollary 1 however:

Proposition 1. *Let $t > 1$ and let w be a t-heterogeneous probability function on L (not purely unary) satisfying Sx. Then w is not a member of any language invariant family all satisfying Sx.*

Proof. Suppose that w is a t-heterogeneous probability function on L and a member of some language invariant family. Let w' be a member of this family on $\mathcal{L} = L \cup \{P_1, P_2, \ldots, P_{t+1}\}$ where the P_i are new unary predicates.

Since L contains a non-unary relation we can find a state description $\Theta(a_1, a_2, \ldots, a_{t+1})$ for $\mathcal{L}$ whose restriction $\Theta_L(a_1, a_2, \ldots, a_{t+1})$ to L has a spectrum of length $t+1$ (so all the $a_1, ..., a_{t+1}$ are mutually distinguishable already in L). If $w'(\Theta(a_1, a_2, \ldots, a_{t+1})) > 0$ then $w(\Theta_L(a_1, a_2, \ldots, a_{t+1})) > 0$, contradicting t-heterogeneity. Hence by Sx $w'(\Psi(a_1, a_2, \ldots, a_{t+1})) = 0$ whenever $\Psi(a_1, a_2, \ldots, a_{t+1})$ has spectrum of length $t+1$ and consequently $w'(\Phi(\boldsymbol{a})) = 0$ whenever the state description $\Phi(\boldsymbol{a})$ has spectrum of length greater than t. (Since any such $\Phi(\boldsymbol{a})$ extends some $\Psi(a_1, a_2, \ldots, a_{t+1})$ with spectrum of length $t+1$).

Now let $\Theta(a_1, a_2, \ldots, a_t)$ be a state description for $\mathcal{L}$ with spectrum of length t whose restriction $\Theta_L(a_1, a_2, \ldots, a_t)$ to L has a spectrum of length 1. Notice that

[6] A similar representation theorem can be proved for t-heterogeneous probability functions, see [16], but that will not be needed here.

since for any spectrum of length t we can find an extension of $\Theta(a_1, a_2, \ldots, a_t)$ with that spectrum it must, by t-heterogeneity be the case that $w'(\Theta(a_1, a_2, \ldots, a_t)) > 0$. Furthermore if $\Phi(a_1, a_2, \ldots, a_{t+j})$ is a state description for $a_1, a_2, \ldots, a_{t+j}$ in $\mathcal{L}$ extending $\Theta(a_1, a_2, \ldots, a_t)$ and $w'(\Phi(a_1, a_2, \ldots, a_{t+j})) \neq 0$ it must be the case that $\Phi(a_1, a_2, \ldots, a_{t+j})$ has spectrum of length t and in consequence the restriction $\Phi_L(a_1, a_2, \ldots, a_{t+j})$ to L must still have spectrum of length 1. Hence for $\Psi_L(a_1, a_2, \ldots, a_r)$ ranging over state descriptions for L and $r > t$,

$$\begin{aligned}
\sum_{|\mathcal{S}(\Psi_L(a_1,\ldots,a_r))|=1} w(\Psi_L(a_1, \ldots, a_r)) &\geq \sum_{\substack{\Phi_L(a_1,\ldots,a_r) \text{ extends } \Theta_L(a_1,\ldots,a_t) \\ |\mathcal{S}(\Phi_L(a_1,\ldots,a_r))|=1}} w(\Phi_L(\boldsymbol{a})) \\
&= \sum_{\substack{\Phi_L(a_1,\ldots,a_r) \text{ extends } \Theta_L(a_1,\ldots,a_t) \\ |\mathcal{S}(\Phi_L(a_1,\ldots,a_r))|=1}} w'(\Phi_L(\boldsymbol{a})) \\
&\geq \sum_{\Phi(a_1,\ldots,a_r) \text{ extends } \Theta(a_1,\ldots,a_t)} w'(\Phi(\boldsymbol{a})) \\
&= w'(\Theta(a_1, \ldots, a_t)).
\end{aligned}$$

Hence

$$\lim_{r\to\infty} \sum_{|\mathcal{S}(\Psi_L(a_1,\ldots,a_r))|=1} w(\Psi_L(a_1, \ldots, a_r)) \geq w'(\Theta(a_1, \ldots, a_t)) > 0,$$

contradicting t-heterogeneity.

Notice however that it is certainly possible to have mixtures of t-heterogeneous probability functions (for different t) which are language invariant. For example if we take $\overline{p} = \langle p_0, p_1, p_2, \ldots \rangle \in \mathbb{B}$ with $p_0 = 0$ and $p_s = 0$ for $s > t$ then $u_L^{\overline{p}}$ is a convex combination of r-heterogeneous probability functions for $r \leq t$ and *is* language invariant by Theorem 1.

Proposition 1 does not hold if $t = 1$, the trivial probability function on L which gives probability 1 to all the a_i being indistinguishable (i.e. $1, 2, 3, \ldots$ all being in the same equivalence class) provides, as L varies, the example of such a language invariant family.

We finally observe that the requirement in Proposition 1 that L contains a non-unary relation can be dropped if $t < 2^s$ where s is the number of unary relation symbols in L.

4 Conclusion

Since both Sx and Language Invariance are (we would claim) desirable principles in the context of assigning beliefs in the absence of any prior knowledge it is pleasing to have a sufficiency theorem for such probability functions in terms of the particularly simple functions $u_L^{\overline{p}}$. This furthermore opens the possibility of deriving certain other properties of such functions by moving the onus of the task onto the much more malleable $u_L^{\overline{p}}$, examples of which will be given in the forthcoming [12].

Acknowledgments. We would like to thank the referees for their helpful comments.

References

1. Carnap, R.: Logical Foundations of Probability, University of Chicago Press, Chicago, Routledge & Kegan Paul Ltd (1950)
2. Carnap, R.: The Continuum of Inductive Methods. University of Chicago Press, Chicago (1952)
3. Carnap, R.: A basic system of inductive logic. In: Jeffrey, R.C. (ed.) Studies in Inductive Logic and Probability, vol. II, pp. 7–155. University of California Press (1980)
4. Gaifman, H.: Concerning measures on first order calculi. Israel journal of mathematics 2, 1–18 (1964)
5. Hill, M.J., Paris, J.B., Wilmers, G.M.: Some observations on induction in predicate probabilistic reasoning. Journal of Philosophical Logic 31(1), 43–75 (2002)
6. Kallenberg, O.: Probabilistic Symmetries and Invariance Principles. Springer, New York (2005)
7. Hoover, D.N.: Relations on probability spaces and arrays of random variables. Institute of Advanced Study, Princeton (1979) (preprint)
8. Johnson, W.E.: Probability: The deductive and inductive problems. Mind 41(164), 409–423 (1932)
9. Kemeny, J.G.: Carnap's theory of probability and induction. In: Schilpp, P.A. (ed.) The Philosophy of Rudolf Carnap, La Salle, Illinois, Open Court, pp. 711–738 (1963)
10. Krauss, P.H.: Representation of symmetric probability models. Journal of Symbolic Logic 34(2), 183–193 (1969)
11. Landes, J.: Doctorial Thesis, Manchester University, UK (to appear)
12. Landes, J., Paris, J.B., Vencovská, A.: Representation Theorems for probability functions satisfying Spectrum Exchangeability in Inductive Logic. Journal of Symbolic Logic (to be submitted)
13. Nix, C.J., Paris, J.B.: A Continuum of inductive methods arising from a generalized principle of instantial relevance. Journal of Philosophical Logic, Online First Issue (2005) (Paper) 1573-0433, ISSN 0022-3611
14. Nix, C.J.: Probabilistic Induction in the Predicate Calculus, Doctorial Thesis, Manchester University, Manchester (2005), See `http://www.maths.man.ac.uk/~jeff/#students`
15. Nix, C.J., Paris, J.B.: A note on binary inductive logic. Journal of Philosophical Logic (to appear)
16. Paris, J.B., Vencovská, A.: From unary to binary inductive logic. In: The Proceedings of the Second Indian Conference on Logic and its Applications, Mumbai (to appear)
17. Paris, J.B.: The Uncertain Reasoner's Companion. Cambridge University Press, Cambridge (1994)
18. Vencovská, A.: Binary Induction and Carnap's Continuum. In: Proceedings of the 7th Workshop on Uncertainty Processing (WUPES), Mikulov (2006), See `http://mtr.utia.cas.cz/wupes06/articles/data/vencovska.pdf`

Best Approximation of Ruspini Partitions in Gödel Logic

Pietro Codara[1], Ottavio M. D'Antona[2], and Vincenzo Marra[2]

[1] Dipartimento di Matematica F. Enriques, Università degli Studi di Milano, via Saldini 50, I-20133 Milano, Italy
codara@mat.unimi.it
[2] Dipartimento di Informatica e Comunicazione, Università degli Studi di Milano, via Comelico 39, I-20135 Milano, Italy
{dantona,marra}@dico.unimi.it

Abstract. A *Ruspini partition* is a finite family of fuzzy sets $\{f_1, \ldots, f_n\}$, $f_i : [0,1] \to [0,1]$, such that $\sum_{i=1}^{n} f_i(x) = 1$ for all $x \in [0,1]$. We analyze such partitions in the language of Gödel logic. Our main result identifies the precise degree to which the Ruspini condition is expressible in this language, and yields *inter alia* a constructive procedure to axiomatize a given Ruspini partition by a theory in Gödel logic.

1 Introduction

Let $[0,1]$ be the real unit interval. By a *fuzzy set* we shall mean a function $f\colon [0,1] \to [0,1]$. Throughout the paper, we fix a finite nonempty family

$$P = \{f_1, \ldots, f_n\}$$

of fuzzy sets, for $n \geq 1$ an integer. Moreover, we write $\underline{n}$ for the set $\{1, \ldots, n\}$.

In several soft computing applications, the following notion of fuzzy partition plays an important role. It is often traced back to [9, p. 28].

Definition 1. *We say P is a* Ruspini partition *if for all $x \in [0,1]$*

$$\sum_{i=1}^{n} f_i(x) = 1\,. \tag{1}$$

By way of informal motivation for what follows, think of the real unit interval $[0,1]$ as the normalized range of values of a physical observable, say "Temperature". Then each $f_i \in P$ can be viewed as a means of assigning a truth-value to a proposition about temperature in some many-valued logic $\mathscr{L}$. Had one no information at all about such propositions, one would be led to identify them with propositional variables X_i, subject only to the axioms of $\mathscr{L}$. However, the set P does encode information about $X_1, \ldots, X_n$. For example, consider $P = \{f_1, f_2, f_3\}$ as in Fig. 1, and say f_1, f_2, and f_3 provide truth-values for the propositions X_1 = "The temperature is low", X_2 = "The temperature is

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 161–172, 2007.

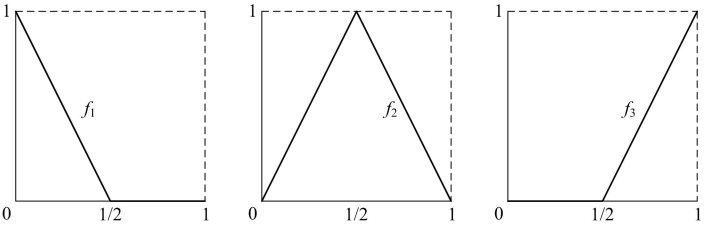


Fig. 1. A Ruspini partition $\{f_1, f_2, f_3\}$

medium", and X_3 = "The temperature is high", respectively. If $\mathscr{L}$ has a conjunction $\wedge$ interpreted by minimum, the proposition $X_1 \wedge X_3$ has 0 as its only possible truth-value, *i.e.*, it is a contradiction. The set P then leads one to add extra-logical axioms to $\mathscr{L}$, *e.g.* $\neg(X_1 \wedge X_3)$, in an attempt to express the fact that one cannot observe both a high and a low temperature at the same time. More generally, P implicitly encodes a *theory* over the pure logic $\mathscr{L}$.

Throughout the paper, we shall take $\mathscr{L}$ to be Gödel logic. Recall that *Gödel* (*propositional*) *logic* can be defined as the schematic extension of the intuitionistic propositional calculus by the prelinearity axiom $(\alpha \to \beta) \vee (\beta \to \alpha)$. It can also be semantically defined as a many-valued logic, as follows. Let us consider well-formed formulas over propositional variables $X_1, X_2, \ldots$ in the language $\wedge, \vee, \to, \neg, \bot, \top$. (We use $\bot$ and $\top$ as the logical constants *falsum* and *verum*, respectively). By an *assignment* we shall mean a function μ from (well-formed) formulas to $[0,1] \subseteq \mathbb{R}$ such that, for any two such formulas $\alpha,\ \beta$,

$$\begin{aligned}
&\mu(\alpha \wedge \beta) = \min\{\mu(\alpha), \mu(\beta)\}\\
&\mu(\alpha \vee \beta) = \max\{\mu(\alpha), \mu(\beta)\}\\
&\mu(\alpha \to \beta) = \begin{cases} 1 & \text{if } \mu(\alpha) \leq \mu(\beta)\\ \mu(\beta) & \text{otherwise}\end{cases}
\end{aligned}$$

and $\mu(\neg\alpha) = \mu(\alpha \to \bot)$, $\mu(\bot) = 0$, $\mu(\top) = 1$. A *tautology* is a formula α such that $\mu(\alpha) = 1$ for every assignment μ. As is well-known, Gödel logic is complete with respect to this many-valued semantics. We refer to *e.g.* [6] [7] for detailed treatments.

This paper provides a thorough analysis of how the Ruspini condition on P is reflected by the resulting theory over Gödel logic. In our main Theorem, we shall eventually obtain a constructive procedure to axiomatize the theory implicitly encoded by P. While it is to be expected that Gödel logic cannot precisely capture addition of real numbers in the Ruspini condition (1), our main result proves that, up to logical equivalence, (1) reduces to the notion of *weak Ruspini partition* in Definition 7. In Section 2 we collect the necessary algebraic and combinatorial background, and prove some preliminary results. In Section 3 we establish our main result. The final Section 4 summarizes our findings.

2 Preliminary Results

We make use of the algebraic counterpart of Gödel logic, namely, the variety of *Gödel algebras*. These are Heyting algebras $\langle G, \wedge, \vee, \neg, \top, \bot \rangle$ satisfying the prelinearity condition $(x \to y) \vee (y \to x) = \top$.

The collection of all functions from $[0,1]$ to $[0,1]$ has the structure of a Gödel algebra under the following operations, for $f, g : [0,1] \to [0,1]$.

$$
\begin{aligned}
(f \wedge g)(x) &= \min\{f(x), g(x)\} \\
(f \vee g)(x) &= \max\{f(x), g(x)\} \\
(f \to g)(x) &= \begin{cases} 1 & \text{if } f(x) \leq g(x) \\ g(x) & \text{otherwise} \end{cases} \\
(\neg f)(x) &= \begin{cases} 1 & \text{if } f(x) = 0 \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}
$$

The top and bottom elements of the algebra are the constant functions 1 and 0, respectively.

We shall denote by $\mathscr{G}(P)$ the Gödel subalgebra of the algebra of all functions from $[0,1]$ to itself generated by P. For each integer $k \geq 0$, we write $\mathscr{G}_k$ for the free Gödel algebra on k free generators $X_1, \ldots, X_k$. Note that, since the variety of Gödel algebras is locally finite, $\mathscr{G}_k$ is finite. Since $\mathscr{G}(P)$ is generated by n elements, there is a congruence Θ on $\mathscr{G}_n$ such that

$$\mathscr{G}_n/\Theta \cong \mathscr{G}(P) \ , \tag{2}$$

where $\cong$ denotes isomorphism of Gödel algebras. Congruences of finite Gödel algebras are principal, so that Θ is generated by a single equation $\alpha(X_1, \ldots, X_n) = \top$ in the language of Gödel algebras. In logical terms, there is a single formula

$$\alpha_P \equiv \alpha_P(X_1, \ldots, X_n) \tag{3}$$

over the n variables $X_1, \ldots, X_n$, such that the Lindenbaum algebra of the theory axiomatized by the single axiom α_P is isomorphic to $\mathscr{G}(P)$. Note that α_P is uniquely determined by P up to logical equivalence. Intuitively, α_P encodes all relations between the fuzzy sets $f_1, \ldots, f_n$ that Gödel logic is capable to express.

In this section, we shall show how to obtain an explicit combinatorial representation of the algebra $\mathscr{G}(P)$ in terms of certain partially ordered sets (*posets*, for short). Recall that, given a poset $(F, \leq)$ and a set $Q \subseteq F$, the *downset* of Q is

$$\downarrow Q = \{x \in F \mid x \leq q, \text{ for some } q \in Q\}.$$

We write $\downarrow q$ for $\downarrow \{q\}$. A poset F is a *forest* if for all $q \in F$ the downset $\downarrow q$ is a chain (*i.e.*, a totally ordered set). A *leaf* is a maximal element of F. A *tree* is a forest with a bottom element, called the *root* of the tree. A *subforest* of a forest F is the downset of some $Q \subseteq F$. The *height* of a chain is the number of its elements. The *height* of a forest is the height of an inclusion-maximal chain of the forest.

Let $\mathrm{Sub}(F)$ denote the family of all subforests of a forest F. Notice that $\mathrm{Sub}(F)$ has a natural structure of Gödel algebra, where $\wedge$ and $\vee$ are given by union and intersection of subforests, and implication is defined, for $F_1, F_2 \in \mathrm{Sub}(F)$, as

$$F_1 \to F_2 = \{q \in F \mid \downarrow q \cap F_1 \subseteq \downarrow q \cap F_2\}.$$

The constants $\bot, \top$ are the empty forest and F itself, respectively. Finally, negation is defined by $\neg F_1 = F_1 \to \bot$.

We introduce a specific forest built from assignments that plays a key role in the following.

Definition 2. *We say that two assignments μ and ν are* equivalent over the first n variables*, or* n-equivalent*, written $\mu \equiv_n \nu$, if and only if there exists a permutation $\sigma : \underline{n} \to \underline{n}$ such that*:

$$0 \preceq_0 \mu(X_{\sigma(1)}) \preceq_1 \cdots \preceq_{n-1} \mu(X_{\sigma(n)}) \preceq_n 1 \ , \tag{4}$$

$$0 \preceq_0 \nu(X_{\sigma(1)}) \preceq_1 \cdots \preceq_{n-1} \nu(X_{\sigma(n)}) \preceq_n 1 \ ,$$

where $\preceq_i \in \{<, =\}$, for $i = 0, \ldots, n$.

Clearly, $\equiv_n$ is an equivalence relation. Throughout, we write $\mathscr{F}_n$ for the (finite) set of equivalence classes of $\equiv_n$.

It is not difficult to show that if $\alpha(X_1, \ldots, X_n)$ is a well-formed formula in Gödel logic, and μ, ν are two n-equivalent assignments, then

$$\mu(\alpha(X_1, \ldots, X_n)) = 1 \ \text{ if and only if } \ \nu(\alpha(X_1, \ldots, X_n)) = 1. \tag{5}$$

We can further endow $\mathscr{F}_n$ with a partial order.

Definition 3. *Let $[\mu]_{\equiv_n}, [\nu]_{\equiv_n} \in \mathscr{F}_n$, and let $\sigma : \underline{n} \to \underline{n}$ be a permutation such that*

$$0 \preceq_0 \nu(X_{\sigma(1)}) \preceq_1 \cdots \preceq_{n-1} \nu(X_{\sigma(n)}) \preceq_n 1 \, ,$$

$$0 \mathrel{\widetilde{\preceq}}_0 \mu(X_{\sigma(1)}) \mathrel{\widetilde{\preceq}}_1 \cdots \mathrel{\widetilde{\preceq}}_{n-1} \mu(X_{\sigma(n)}) \mathrel{\widetilde{\preceq}}_n 1 \, ,$$

where $\preceq_i$, $\mathrel{\widetilde{\preceq}}_i \ \in \{<, =\}$, for $i = 0, \ldots, n$. We define $[\mu]_{\equiv_n} \leq [\nu]_{\equiv_n}$ if and only if there exists an index $k \in \{0, \ldots, n\}$ such that

i) $\mathrel{\widetilde{\preceq}}_i$ coincides with $\preceq_i$ if $0 \leq i \leq k$,

ii) $\mathrel{\widetilde{\preceq}}_i$ coincides with $=$ if $k+1 \leq i \leq n$.

Example 1. Let μ, ν, ξ be assignments such that

- $\mu(X_1) = 1$, $\mu(X_2) = 1/3$, $\mu(X_3) = 0$, $\mu(X_4) = 1$,
- $\nu(X_1) = 1$, $\nu(X_2) = 1/4$, $\nu(X_3) = 0$, $\nu(X_4) = 1/2$,
- $\xi(X_1) = 1$, $\xi(X_2) = 1/2$, $\xi(X_3) = 0$, $\xi(X_4) = 1/2$.

For $\sigma(1) = 3$, $\sigma(2) = 2$, $\sigma(3) = 4$, $\sigma(4) = 1$, one has

- $0 = \mu(X_3) < \mu(X_2) < \mu(X_4) = \mu(X_1) = 1$,

– $0 = \nu(X_3) < \nu(X_2) < \nu(X_4) < \nu(X_1) = 1$,
– $0 = \xi(X_3) < \xi(X_2) = \xi(X_4) < \xi(X_1) = 1$.

Thus, according to Definition 3, $[\mu]_{\equiv_n} \leq [\nu]_{\equiv_n}$, and $[\xi]_{\equiv_n}$ is uncomparable to both $[\mu]_{\equiv_n}$ and $[\nu]_{\equiv_n}$.

One checks that $\leq$ in Definition 3 indeed is a partial order on $\mathscr{F}_n$, and $(\mathscr{F}_n, \leq)$ is in fact a forest [4, Lemma 3.3]. We immediately notice that

a) the roots of the trees are the classes of Boolean assignments,
b) the class $[\mu]_{\equiv_n}$ such that $\mu(X_1) = \cdots = \mu(X_n) = 0$ is the only tree having height 1, and
c) the leaves are those classes of assignments in which no variable is set to 1.

For each $i = 1, \ldots, n$, let $\chi_i = \{[\mu]_{\equiv_n} \mid \mu(X_i) = 1\}$ be the i^{th} *generating subforest* of $\mathscr{F}_n$.

Proposition 1. *Fix an integer $k \geq 0$. (i) $\mathrm{Sub}(\mathscr{F}_k)$ is (isomorphic to) the free Gödel algebra on k free generators. A free generating set is given by the collection of generating subforests. (ii) Up to isomorphism, the quotients of $\mathrm{Sub}(\mathscr{F}_k)$ are precisely the algebras of the form $\mathrm{Sub}(F)$, for $F \in \mathrm{Sub}(\mathscr{F}_k)$. (iii) The set of prime filters ordered by reverse inclusion of $\mathrm{Sub}(F)$ is order-isomorphic to F for every $F \in \mathrm{Sub}(\mathscr{F}_k)$.*

Proof. The proof is a straightforward translation of [5, Remark 2 and Proposition 2.4] in the language of assignments. □

Figure 2 shows the forest $\mathscr{F}_2$, whose nodes are labelled by the ordering of variables under a given assignment as in (4). However, for the sake of readability, here and in the following figure we write X_i instead of $\mu(X_i)$.

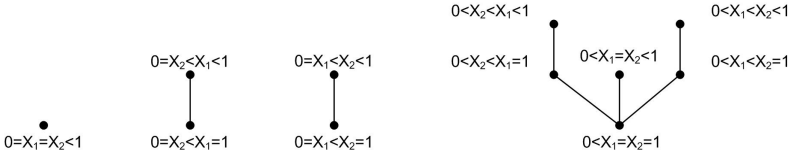


Fig. 2. The forest $\mathscr{F}_2$

As an immediate consequence of Proposition 1, we can reformulate (2) as follows: P uniquely determines a congruence Θ' on $\mathrm{Sub}(\mathscr{F}_n)$, and a subforest $\mathscr{F}(P)$ of $\mathscr{F}_n$ such that

$$\mathrm{Sub}(\mathscr{F}_n)/\Theta' \cong \mathrm{Sub}(\mathscr{F}(P)) \cong \mathscr{G}(P)$$

To relate Θ' with the formula α_P in (3) or, equivalently, with $\mathscr{F}(P)$, we shall give an explicit description of $\mathscr{F}(P)$. To this end, it is convenient to introduce the following notion.

Definition 4. *Let* $[\mu]_{\equiv_n} \in \mathscr{F}_n$ *and* $x \in [0,1]$. *We say* $[\mu]_{\equiv_n}$ *is* realized by P at x *if there exists a permutation* $\sigma : \underline{n} \to \underline{n}$ *such that*

$$0 \preceq_0 f_{\sigma(1)}(x) \preceq_1 \cdots \preceq_{n-1} f_{\sigma(n)}(x) \preceq_n 1\,,$$

$$0 \preceq_0 \mu(X_{\sigma(1)}) \preceq_1 \cdots \preceq_{n-1} \mu(X_{\sigma(n)}) \preceq_n 1\,,$$

where $\preceq_i \in \{<, =\}$, $i \in \{0, \ldots, n\}$.

Proposition 2.

$$\mathscr{F}(P) = \downarrow \{[\mu]_{\equiv_n} \in \mathscr{F}_n \mid [\mu]_{\equiv_n} \textit{ is realized by } P \textit{ at some } x \in [0,1]\}$$

Proof. We construct a subdirect representation of $\mathscr{G}(P)$ as follows. There exists a finite set $\{x_1, \ldots, x_m\} \subseteq [0,1]$ such that for each $y \in [0,1]$, if $[\mu]_{\equiv_n} \in \mathscr{F}(P)$ is realized by P at y, then it is also realized by P at x_i, for some $i \in \underline{m}$. Moreover, one checks that evaluating the elements of $\mathscr{G}(P)$ at x_i yields a totally ordered Gödel algebra C_{x_i} that is a homomorphic image of $\mathscr{G}(P)$ via the quotient map q_i given by restriction to x_i. The homomorphism

$$s : \mathscr{G}(P) \hookrightarrow \prod_{i=1}^{m} C_{x_i}$$

given by

$$g \in \mathscr{G}(P) \longmapsto (q_1(g), \ldots, q_m(g))$$

is injective. Indeed, let $g \neq h \in \mathscr{G}(P)$, say $g(y) > h(y)$ for $y \in [0,1]$. For the sake of brevity, we shall only deal with the case $1 > g(y) > h(y) > 0$. Then $g(y) = f_i(y)$ and $h(y) = f_j(y)$ for $i \neq j$. Let $[\mu]_{\equiv_n}$ be the assignment realized by P at y. There exists $u \in \underline{m}$ such that $[\mu]_{\equiv_n}$ is realized by P at x_u, and therefore $f_i(x_u) > f_j(x_u)$, which proves $s(g) \neq s(h)$.

It now follows that s is a subdirect representation of $\mathscr{G}(P)$. By Proposition 1(*iii*) we identify prime filters of $\mathscr{G}(P)$ with elements of $\mathscr{F}(P) \subseteq \mathscr{F}_n$. The primes that are kernels of $q_1, \ldots, q_m$ must comprise all inclusion-minimal primes of $\mathscr{G}(P)$, *i.e.*, all leaves of $\mathscr{F}(P)$, for otherwise s could not be a subdirect representation. Therefore, the classes $[\mu]_{\equiv_n}$ realized by P at some $x \in [0,1]$ comprise all leaves of $\mathscr{F}(P)$ (and possibly other elements). Since any forest is the downset of its leaves the proposition is proved. □

Moreover, we associate with a formula $\alpha(X_1, \ldots, X_n)$ the uniquely determined subforest of $\mathscr{F}_n$, denoted $\mathscr{F}_\alpha$, as follows:

$$\mathscr{F}_\alpha = \{[\mu]_{\equiv_n} \in \mathscr{F}_n \mid \mu(\alpha) = 1\}\,.$$

By (5), $\mathscr{F}_\alpha$ does not depend on the choice of μ. Clearly, $\mathscr{F}_\alpha$ corresponds to the quotient algebra $\mathrm{Sub}(\mathscr{F}_n)/\Theta'$, where Θ' is the congruence generated by $\alpha(X_1, \ldots, X_n) = \top$. Finally, by the foregoing we have

$$\mathscr{F}_{\alpha_P} = \mathscr{F}(P)\,. \tag{6}$$

3 Gödel Approximation of Ruspini Partitions

Definition 5. *We denote by $\mathscr{R}_n$ the subforest of $\mathscr{F}_n$ obtained by removing from $\mathscr{F}_n$ the single tree having height* 1*, and the leaves of all the trees having height* 2*. We call $\mathscr{R}_n$ the* Ruspini forest*.*

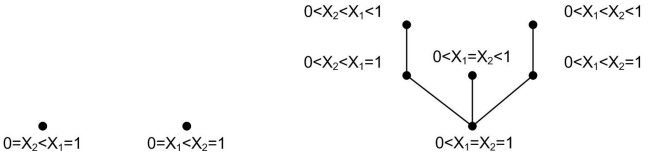


Fig. 3. The Ruspini forest $\mathscr{R}_2$

We now show how to explicitly axiomatize $\mathscr{R}_n$.

Definition 6. *We define the* Ruspini axiom $\rho_n = \alpha \vee \beta$*, where*

$$\alpha = \bigvee_{1 \le i < j \le n} (\neg\neg X_i \wedge \neg\neg X_j)\,, \quad \textit{and} \quad \beta = \bigvee_{1 \le i \le n} (X_i \wedge \bigwedge_{1 \le j \ne i \le n} \neg X_j)\,.$$

Lemma 1. $\mathscr{F}_{\rho_n} = \mathscr{R}_n$.

Proof. Fix an assignment μ. Since

$$\mu(\neg\neg X) = \begin{cases} 0 & \text{if } \mu(X) = 0 \\ 1 & \text{otherwise,} \end{cases}$$

$\mu(\alpha) \ne 1$ if and only if at most one variable X_{i_0} satisfies $\mu(X_{i_0}) \ne 0$.

Observe now that $\mu(\beta) = 1$ if and only if there exists $i \in \underline{n}$ such that, for $j \ne i$, $\mu(X_i) = 1$ and $\mu(X_j) = 0$.

Therefore, $\mu(\rho_n) = \mu(\alpha \vee \beta) \ne 1$ if and only if there exists $i_0 \in \underline{n}$ such that, for $j \ne i_0$, $\mu(X_{i_0}) < 1$ and $\mu(X_j) = 0$. It is now straightforward to verify that the latter condition holds if and only if $[\mu]_{\equiv_n} \notin \mathscr{R}_n$. □

Let us introduce a property of P that we shall use in our main result. Let $\lambda : [0,1] \to [0,1]$ be an order preserving map such that $\lambda(0) = 0$ and $\lambda(1) = 1$, and let $t = \inf \lambda^{-1}(1)$. If the restriction of λ to $[0,t]$ is an order isomorphism between $[0,t]$ and $[0,1]$, we say λ is a *comparison map*.

Definition 7. *We say P is a* weak Ruspini partition *if for all $x \in [0,1]$, there exist $y \in [0,1]$, a comparison map λ, and an order isomorphism $\gamma : [0,1] \to [0,1]$ such that*

(i) $\lambda(f_i(y)) = f_i(x)$*, for all $i \in \underline{n}$.*
(ii) $\sum_{i=1}^{n} \gamma(f_i(y)) = 1$.

Lemma 2. *Let $[\mu]_{\equiv_n}$, $[\nu]_{\equiv_n} \in \mathscr{F}_n$ and $x, y \in [0,1]$ such that $[\mu]_{\equiv_n}$ and $[\nu]_{\equiv_n}$ are realized by P at x and y, respectively. Then the following are equivalent.*

(*i*) $[\mu]_{\equiv_n} \leq [\nu]_{\equiv_n}$.
(*ii*) *There exists a comparison map $\lambda : [0,1] \to [0,1]$ with $\lambda(f_i(y)) = f_i(x)$, for all $i \in \underline{n}$.*

Moreover, the following are equivalent.

(*iii*) $[\mu]_{\equiv_n}$ *is a leaf of $\mathscr{R}_n$.*
(*iv*) *There exists an order isomorphism $\gamma : [0,1] \to [0,1]$ with $\sum_{i=1}^{n} \gamma(f_i(x)) = 1$.*

Proof. (*i*)⇒(*ii*). By Definitions 3 and 4, there exists a permutation $\sigma : \underline{n} \to \underline{n}$ such that

$$0 \preceq_0 f_{\sigma(1)}(y) \preceq_1 \cdots \preceq_{n-1} f_{\sigma(n)}(y) \preceq_n 1\,,$$

$$0 \mathrel{\widetilde{\preceq}}_0 f_{\sigma(1)}(x) \mathrel{\widetilde{\preceq}}_1 \cdots \mathrel{\widetilde{\preceq}}_{n-1} f_{\sigma(n)}(x) \mathrel{\widetilde{\preceq}}_n 1\,,$$

where $\preceq_i, \mathrel{\widetilde{\preceq}}_i \in \{<, =\}$, and there is $k \in \{0, \ldots, n\}$ satisfying *i*) and *ii*) in Definition 3. We deal with the case $k < n$ only; the case $k = n$ is a trivial variation thereof. We define Λ by $\Lambda(f_{\sigma(i)}(y)) = f_{\sigma(i)}(x)$, for $1 \leq i \leq k$, and $\Lambda(f_{\sigma(i)}(y)) = 1$ if $k+1 \leq i \leq n$. We extend Λ to a comparison map as follows. Consider the closed intervals $I_0 = [0, f_{\sigma(1)}(y)]$, $J_0 = [0, f_{\sigma(1)}(x)]$, $I_i = [f_{\sigma(i)}(y), f_{\sigma(i+1)}(y)]$ and $J_i = [f_{\sigma(i)}(x), f_{\sigma(i+1)}(x)]$, for $1 \leq i \leq k$. Now let us fix $0 \leq h \leq k$. Note that if I_h collapses to a point, then J_h also collapses to a point. Therefore in all cases we can choose order isomorphisms $\lambda_h : I_h \to J_h$. Moreover, set $I_{k+1} = [f_{\sigma(k+1)}(y), 1]$ and $\lambda_{k+1} : I_{k+1} \to \{1\}$. Since λ_h and λ_{h+1} agree at $I_h \cap I_{h+1}$ by construction, the function $\lambda : [0,1] \to [0,1]$ defined by $\lambda(r) = \lambda_j(r)$ if $r \in I_j$, for $0 \leq j \leq k+1$, is a comparison map satisfying (*ii*).

(*ii*)⇒(*i*). Immediate from Definitions 3 and 4.

(*iii*)⇒(*iv*). It is an exercise to check that $[\mu]_{\equiv_n}$ is a leaf of $\mathscr{R}_n$ if and only if exactly one of the following two cases hold.

Case 1. There exists i_0 such that $\mu(X_{i_0}) = 1$ and $\mu(X_i) = 0$ for $i \neq i_0$.

Let γ be the identity map. By Definition 4, we have $\sum_{i=1}^{n} \gamma(f_i(x)) = 1$.

Case 2. For all i, $\mu(X_i) < 1$, and there exist i_0, i_1 such that $0 < \mu(X_{i_0}) \leq \mu(X_{i_1})$. Let us write

$$0 \preceq_0 f_{\sigma(1)}(x) \preceq_1 \cdots \preceq_{n-1} f_{\sigma(n)}(x) \preceq_n 1\,,$$

for some permutation σ and $\preceq_i \in \{<, =\}$. We shall assume $\preceq_0$ is $<$. The case where some f_i takes value zero at x is entirely similar.

Now consider the $(n-1)$-dimensional simplex[1] S_n, given by the convex hull of the standard basis of $\mathbb{R}_n$. Let $S_n^{(1)}$ be the simplicial complex given by the first barycentric subdivision of S_n. The $(n-1)$-dimensional simplices of $S_n^{(1)}$ are in bijection with permutations of $\underline{n}$, and the solution set of the inequalities

$$0 \leq r_1 \leq \cdots \leq r_n \leq 1 \tag{7}$$

[1] For all unexplained notions in combinatorial topology, please see [8].

in S^n is an $(n-1)$-dimensional simplex $S \in S_n^{(1)}$. Consider the equalities

$$r_i = r_{i+1} \tag{8}$$

for each $i = 1, \ldots, n-1$ such that $\preceq_i$ is $=$. Then the solution set of (7) and (8) is a nonempty face T of S. Consider next the strict inequalities

$$\begin{cases} r_i < r_{i+1} \\ 0 < r_1 \\ r_n < 1 \end{cases} \tag{9}$$

for all $i = 1, \ldots, n-1$ such that $\preceq_i$ is $<$. Then the solution set of (7), (8), and (9) is the relative interior T° of T. Since T is nonempty, T° is nonempty. The barycenter $b = (b_1, \ldots, b_n)$ of T lies in T°. Since $b \in S_n$, we have $\sum_{k=1}^n b_k$. Moreover, by construction,

$$0 \preceq_0 b_1 \preceq_1 \cdots \preceq_{n-1} b_n \preceq_n 1\,.$$

We define Γ by $\Gamma(f_{\sigma(i)}) = b_i$. Arguing as in the proof of $(i) \Rightarrow (ii)$, we conclude that there is an extension of Γ to an order isomorphism $\gamma : [0,1] \to [0,1]$ satisfying (iv).

$(iv) \Rightarrow (iii)$. Suppose $[\mu]_{\equiv_n}$ is not a leaf of $\mathscr{R}_n$. Thus, exactly one of the following two cases holds.

Case 1. $[\mu]_{\equiv_n} \in \mathscr{F}_n \setminus \mathscr{R}_n$.

In this case there exists i_0 such that $\mu(X_{i_0}) < 1$ and $\mu(X_i) = 0$ for $i \neq i_0$. Using Definition 4, we have $\sum_{i=1}^n \gamma(f_i(x)) < 1$, for each order isomorphism γ.

Case 2. $[\mu]_{\equiv_n} \in \mathscr{R}_n$, but $[\mu]_{\equiv_n} \in \mathscr{R}_n$ is not a leaf of $\mathscr{R}_n$.

It is easy to check that there exist i_0, i_1 such that $0 < \mu(X_{i_0}) \leq \mu(X_{i_1}) = 1$. Using Definition 4, we have $f_{i_1}(x) = 1$ and $f_{i_0}(x) > 0$, and thus $\sum_{i=1}^n \gamma(f_i(x)) > 1$, for each order isomorphism γ. □

To state our main result we still need to show how to obtain a formula $\psi_{[\mu]_{\equiv_n}}$ associated with a given element $[\mu]_{\equiv_n} \in \mathscr{F}_n$ such that $\psi_{[\mu]_{\equiv_n}}$ evaluates to 1 exactly on $\downarrow [\mu]_{\equiv_n}$. For this, we define the derived connective $\alpha \triangleleft \beta = ((\beta \to \alpha) \to \beta)$. Given an assignement μ we have that

$$\mu(\alpha \triangleleft \beta) = \begin{cases} 1 & \text{if } \mu(\alpha) < \mu(\beta) \text{ or } \mu(\alpha) = \mu(\beta) = 1 \\ \mu(\beta) & \text{otherwise.} \end{cases}$$

Suppose now that, for a given permutation $\sigma : \underline{n} \to \underline{n}$,

$$0 \preceq_0 \mu(X_{\sigma(1)}) \preceq_1 \cdots \preceq_{n-1} \mu(X_{\sigma(n)}) \preceq_n 1\,,$$

where $\preceq_i \in \{<, =\}$, $i = 0, \ldots, n$. We associate to $[\mu]_{\equiv_n}$ the formula

$$\psi_{[\mu]_{\equiv_n}} = (\bot \ \bowtie_0 \ X_{\sigma(1)}) \wedge (X_{\sigma(1)} \ \bowtie_1 \ X_{\sigma(2)}) \wedge \cdots \wedge (X_{\sigma(n)} \ \bowtie_n \ \top)\,,$$

where $\bowtie_i = \triangleleft$ if $\preceq_i$ is $<$, and $\bowtie_i = \leftrightarrow$ otherwise.

Lemma 3. $\mathscr{F}_{\psi_{[\mu]_{\equiv_n}}} = \downarrow [\mu]_{\equiv_n}$

Proof. We omit the straightforward verification. Compare [3,1] where a full-fledged theory of normal forms is developed. □

Given a forest $F \subseteq \mathscr{F}_n$ let us indicate with $\mathrm{Root}(F)$ the set of roots of F (*i.e.* the classes of Boolean assignments over the first n variables). If $r \in \mathrm{Root}(F)$, we write $\mathrm{Leaf}(r, F)$ for the set of leaves of F above the root r.

Definition 8. *We say that a forest F is a* Ruspini subforest *if $F \subseteq \mathscr{R}_n$ and each leaf of F is a leaf of $\mathscr{R}_n$.*

We write $\vdash \alpha$ if Gödel logic proves the formula α; equivalently, by completeness, if $\mu(\alpha) = 1$ for all assignments μ. We can finally prove our main result.

Theorem. *The following are equivalent.*

(*i*) *P is a weak Ruspini partition.*
(*ii*) *$\mathscr{F}(P)$ is a Ruspini subforest.*
(*iii*) $\vdash \alpha \wedge \beta \wedge \gamma$, *where*
$\alpha = (\alpha_P \to \rho_n)$,
$\beta = \bigwedge_{r \in \mathrm{Root}(\mathscr{R}_n)} \bigwedge_{l \in \mathrm{Leaf}(r, \mathscr{R}_n)} \big((\psi_l \to \alpha_P) \vee ((\psi_l \wedge \alpha_P) \to \psi_r)\big)$,
$\gamma = \bigwedge_{r \in \mathrm{Root}(\mathscr{R}_n)} \big((\psi_r \to \alpha_P) \to (\bigvee_{l \in \mathrm{Leaf}(r, \mathscr{R}_n)} (\psi_l \to \alpha_P))\big)$.

Moreover, for any Ruspini subforest F there exists a Ruspini partition $P' = \{f'_1, \ldots, f'_n\}$, with $f'_i : [0,1] \to [0,1]$, such that $\mathscr{F}(P') = F$.

Proof. (*i*) $\Rightarrow$ (*ii*). By Lemma 2, we can reformulate Definition 7 in terms of assignments as follows. For all $[\mu]_{\equiv_n} \in \mathscr{F}_n$ realized by P at some $x \in [0,1]$, there exists $[\nu]_{\equiv_n} \geq [\mu]_{\equiv_n}$ realized by P at some $y \in [0,1]$ such that $[\nu]_{\equiv_n}$ is a leaf of $\mathscr{R}_n$. Thus, by Proposition 2, $\mathscr{F}(P)$ is exactly the downset of those leaves of $\mathscr{R}_n$ realized by P at some $x \in [0,1]$.

(*ii*) $\Rightarrow$ (*iii*). Let $r \in \mathrm{Root}(\mathscr{R}_n)$. If $r \notin \mathscr{F}(P)$ then the set $\mathrm{Leaf}(r, \mathscr{F}(P))$ is empty, and by (6) the formula α_P evaluates to zero under all assignments μ such that $[\mu]_{\equiv_n} \geq r$. Thus, for all $l \in \mathrm{Leaf}(r, \mathscr{R}_n)$, $(\psi_l \wedge \alpha_P)$ is a contradiction and $(\psi_l \wedge \alpha_P) \to \psi_r$ is a tautology. Therefore, the conjucts of β indexed by $r \notin \mathscr{F}(P)$ are tautologies. Moreover, since ψ_r and ψ_l evaluate to zero under all assignments μ such that $[\mu]_{\equiv_n} \not\geq r$, and to a value different from zero otherwise, $(\psi_r \to \alpha_P) \leftrightarrow (\psi_l \to \alpha_P)$ is a tautology for all $l \in \mathrm{Leaf}(r, \mathscr{R}_n)$, and then the conjucts of γ indexed by $r \notin \mathscr{F}(P)$ are tautologies. Let now $r \in \mathscr{F}(P)$, and let $l \in \mathrm{Leaf}(r, \mathscr{R}_n)$. If $l \in \mathscr{F}(P)$ then $\psi_l \to \alpha_P$ is a tautology, otherwise $(\psi_l \wedge \alpha_P) \to \psi_r$ is a tautology. Thus, every formula in β indexed by $r \in \mathscr{F}(P)$ is a tautology. Moreover, $\mathrm{Leaf}(r, \mathscr{F}(P)) \neq \emptyset$, and for $l_0 \in \mathrm{Leaf}(r, \mathscr{F}(P))$, $\psi_{l_0} \to \alpha_P$ is a tautology. Therefore, every formula in γ indexed by $r \in \mathscr{F}(P)$ is a tautology. We thus obtain that β and γ are tautologies. Since α also is a tautology by the hypothesis $\mathscr{F}(P) \subseteq \mathscr{R}_n$, we obtain that whenever (*ii*) holds, $\alpha \wedge \beta \wedge \gamma$ is a tautology.

(*iii*) $\Rightarrow$ (*i*). Suppose P is not a weak Ruspini partition. By Definition 7, using Lemma 2 and (6), there exists $[\mu]_{\equiv_n} \in \mathscr{F}_{\alpha_P}$ such that exactly one of the following two condition hold.

(*a*) $[\mu]_{\equiv_n} \in \mathscr{F}_n \setminus \mathscr{R}_n$.

(*b*) $[\mu]_{\equiv_n} \in \mathscr{R}_n$ is a maximal element of $\mathscr{F}_{\alpha_P}$, but it is not a leaf of $\mathscr{R}_n$.

If (*a*) holds then, clearly, $\mu(\alpha) \neq 1$. As to (*b*), let $r \leq [\mu]_{\equiv_n}$ be a root of $\mathscr{F}_n$. If $r = [\mu]_{\equiv_n}$, then the formula $\psi_r \to \alpha_P$ is a tautology, while $\psi_l \to \alpha_P$ does not evaluate to 1 at l, for all $l \in \mathrm{Leaf}(r, \mathscr{R}_n)$. Thus, γ is not a tautology. If $r \neq [\mu]_{\equiv_n}$ then for all $l \in \mathrm{Leaf}(r, \mathscr{R}_n)$, $l \geq [\mu]_{\equiv_n}$, $\psi_l \to \alpha_P$ and $(\psi_l \wedge \alpha_P) \to \psi_r$ evaluate to zero at l. Therefore β is not a tautology. In any case, $\alpha \wedge \beta \wedge \gamma$ is not a tautology.

Finally, we prove the last statement of the theorem. Let $[\mu_1]_{\equiv_n}, \ldots, [\mu_m]_{\equiv_n}$ be the leaves of F. Partition the interval $[0,1]$ into m intervals $I_1 = [0, x_1]$, $I_2 = (x_1, x_2], \ldots, I_m = (x_{m-1}, 1 = x_m]$. We construct the functions f'_i as follows. For $i \in \underline{n}$, $j \in \underline{m}$, we set $f'_i(x) = C_{ij} \in \mathbb{R}$ if $x \in I_j$. The constants C_{ij} are chosen so that

(*a*) $[\mu_j]_{\equiv_n}$ is realized by P' at x_j,

(*b*) $\sum_{i=1}^{n} C_{ij} = 1$.

Obviously, it is always possible to choose C_{ij} so that (*a*) holds. The proof of (*iii*) $\Rightarrow$ (*iv*) in Lemma 2 shows that, in fact, it is always possible to choose C_{ij} so that both (*a*) and (*b*) hold. □

In [2, Theorem 3] it is shown that the number of leaves of $\mathscr{F}_n$ is

$$L_n = 2 \sum_{k=1}^{n} k! \left\{ {n \atop k} \right\}, \tag{10}$$

where $\left\{ {n \atop k} \right\}$ is the number of partitions of an n-element set into k classes, *i.e.* the *Stirling number of the second kind*. The number $\sum_{k=1}^{n} k! \left\{ {n \atop k} \right\}$ is the n^{th} *ordered Bell number*, *i.e.* the number of all ordered partitions of $\underline{n}$. Compare sequence *A*000670 in [10].

Consider $P' = \{f'_1, \ldots, f'_n\}$, where $f'_i : [0,1] \to [0,1]$. In the light of Section 2, let us say that P' is *Gödel-equivalent* to P if $\mathscr{F}(P) = \mathscr{F}(P')$, or, equivalently, $\vdash \alpha_P \leftrightarrow \alpha_{P'}$. Then:

Corollary 1. *The number of classes of Gödel-equivalent weak Ruspini partitions of n elements is $2^{L_n - 1} - 1$, where L_n is given by* (10).

Proof. A weak Ruspini partition P is characterized, up to Gödel-equivalence, by the forest $\mathscr{F}(P)$, and therefore by a subset of leaves of $\mathscr{R}_n$. Noting that the number of leaves of $\mathscr{R}_n$ is $L_n - 1$, and that for every weak Ruspini partition P, $\mathscr{F}(P) \neq \emptyset$, the corollary follows. □

Corollary 2. (*i*) *There is a Ruspini subforest F such that whenever $\mathscr{F}(P) = F$ then each $f_i \in P$ has a point of discontinuity.* (*ii*) *For all Ruspini subforests F*

with L leaves there is a choice of a Ruspini partition $P' = \{f'_1, \ldots, f'_n\}$, with $\mathscr{F}(P') = F$ such that each $f'_i : [0,1] \to [0,1]$ has at most $L-1$ points of discontinuity.

Proof. (i) It suffices to choose $F \subseteq \mathscr{R}_n$ as the forest of all Boolean assignments which are leaves of $\mathscr{R}_n$. (ii) The construction used in the proof of the last statement of Theorem 3 yields the desired P'. □

4 Conclusions

Our analysis shows that Gödel logic does not have sufficient expressive power to capture the Ruspini condition (1). However, we have proved that Gödel logic does capture the notion of weak Ruspini partition in Definition 7. Moreover, our Theorem 3 shows that weak Ruspini partitions indeed are the best available approximation of Ruspini partitions in Gödel logic: for each weak Ruspini partition P, there exists a Ruspini partition P' that is Gödel-equivalent to P. Thus, there is no formula in Gödel logic telling P and P' apart. Moreover, Corollary 2 shows that one can always choose a Ruspini partition P' whose elements have a bounded number of points of discontinuity. Finally, up to Gödel equivalence, there is a finite number of weak Ruspini partitions of n elements, and Corollary 1 gives an exact formula to compute this number.

References

1. Aguzzoli, S., D'Antona, O.M., Marra, V.: Algorithms in propositional Gödel logic (in preparation)
2. Aguzzoli, S., Gerla, B., Manara, C.: Poset representation for Gödel and Nilpotent Minimum logics. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 662–674. Springer, Heidelberg (2005)
3. Baaz, M., Veith, H.: Interpolation in fuzzy logic. Arch. Math. Logic 38(7), 461–489 (1999)
4. Codara, P., D'Antona, O.M., Marra, V.: Propositional Gödel logic and Delannoy paths. In: Proceedings of Fuzz-IEEE 2007 (to appear)
5. D'Antona, O.M., Marra, V.: Computing coproducts of finitely presented Gödel algebras. Ann. Pure Appl. Logic 142(1-3), 202–211 (2006)
6. Gottwald, S.: A treatise on many-valued logics. In: Studies in Logic and Computation, vol. 9, Research Studies Press Ltd, Baldock (2001)
7. Hájek, P.: Metamathematics of fuzzy logic. In: Trends in Logic—Studia Logica Library, vol. 4, Kluwer Academic Publishers, Dordrecht (1998)
8. Rourke, C.P., Sanderson, B.J.: Introduction to piecewise-linear topology. Springer Study Edition. Springer, Berlin (1982) (reprint)
9. Ruspini, E.H.: A new approach to clustering. Information and Control 15, 22–32 (1969)
10. Sloane, N.J.A.: The on-line encyclopedia of integer sequences (2006), Published electronically at `http://www.research.att.com/~njas/sequences/`

A Logical Approach to Qualitative and Quantitative Reasoning

Emad Saad

College of Computer Science and Information Technology
Abu Dhabi University
Abu Dhabi, UAE
emad.saad@adu.ac.ae

Abstract. Reasoning with qualitative and quantitative uncertainty is required in some real-world applications [6]. However, current extensions to logic programming with uncertainty support representing and reasoning with either qualitative or quantitative uncertainty. In this paper we extend the language of Hybrid Probabilistic Logic programs [29,27], originally introduced for reasoning with quantitative uncertainty, to support both qualitative and quantitative uncertainty. We propose to combine disjunctive logic programs [10,19] with Extended and Normal Hybrid Probabilistic Logic Programs (EHPP [27] and NHPP [29]) in a unified logic programming framework, to allow directly and intuitively to represent and reason in the presence of both qualitative and quantitative uncertainty. The semantics of the proposed languages are based on the answer sets semantics and stable model semantics of extended and normal disjunctive logic programs [10,19]. In addition, they also rely on the probabilistic answer sets semantics and the stable probabilistic model semantics of EHPP [27] and NHPP [29].

1 Introduction

Reasoning under uncertainty is crucial in most real-world applications such as planning with uncertain domains and reasoning about actions with uncertain effects—such as the actions that arises from robotics in real-world environments. The literature is rich with different forms of uncertainty in logic programming. These forms of uncertainty can be classified into qualitative and quantitative models of uncertainty. Qualitative uncertainty is represented in logic programming using disjunctive logic programs [19,10,2]. It often happens that $a \vee b \vee c$ occurs while we are uncertain which of these propositions is true [2]. There might be states of the world where a is true or b is true or c is true or any combinations of them might be also true [2]. On the other hand, quantitative uncertainty is represented in logic programming by means of different formalisms including probability theory (see [28] for a survey on these different formalisms). Probabilistic logic programming is motivated by the need to provide the ability to represent both logical as well as probabilistic knowledge by logic programs (see [29] for a survey on these probabilistic logic programming frameworks).

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 173–186, 2007.

The semantics of such frameworks provide ways to systematically derive logical conclusions along with their associated probabilistic properties.

Among these probabilistic logic programming frameworks is Hybrid Probabilistic Logic Programs (HPP) [28] that modifies the original Hybrid Probabilistic Logic Programming framework of [4], and generalizes and modifies the *probabilistic annotated logic programming framework*, originally proposed in [21] and further extended in [22]. Probabilities in [28] are presented in form of intervals where a probability interval represents the bounds on the degree of belief a rational agent has about the truth of an event. The semantics of HPP [28], intuitively, captures the probabilistic reasoning according to how likely are the various events to occur. It was shown that the HPP [28] framework is more suitable for reasoning and decision making tasks. In addition, it subsumes Lakshmanan and Sadri's [14] probabilistic implication-based framework as well as it is a natural extension of classical logic programming. As a step towards enhancing its reasoning capabilities, the framework of HPP was extended to cope with non-monotonic negation [29] by introducing the notion of Normal Hybrid Probabilistic Logic Programs (NHPP) and providing two different semantics namely; stable probabilistic model semantics and well-founded probabilistic model semantics. Furthermore, NHPP was extended to Extended Hybrid Probabilistic Logic Programs (EHPP) [27] to cope directly with classical negation as well as non-monotonic negation to allow reasoning in the presence of incomplete knowledge. It was shown that Baral et al's probabilistic logic programming approach for reasoning with causal Bayes networks (P-log) [1] is naturally subsumed by EHPP [27].

To this end, disjunctive logic programs are only used for representing and reasoning under qualitative uncertainty and probabilistic logic programming, represented by NHPP and EHPP, are only used for representing and reasoning under quantitative uncertainty. However, in some real-world applications, representing and reasoning with both forms of uncertainty is required [6].

We propose to combine disjunctive logic programs [10,2] with Extended and Normal Hybrid Probabilistic Logic Programs (EHPP [27] and NHPP [29]) in a unified logic programming framework, to allow directly and intuitively to represent and reason in the presence of both qualitative and quantitative uncertainty. This is achieved by introducing the notions of *Extended and Normal Disjunctive Hybrid Probabilistic Logic Programs* (EDHPP and NDHPP). EDHPP and NDHPP generalize extended and normal disjunctive logic programs of classical logic programming [10,2], respectively, as well as, they generalize EHPP and NHPP [27,29]. The semantics of EDHPP and NDHPP are based on the answer sets semantics and stable model semantics of extended and normal disjunctive logic programs [10,2]. The semantics of EDHPP employs the *Open World Assumption*, whereas, the semantics of NDHPP employs the *Closed World Assumption*. We show that EDHPP naturally subsumes extended disjunctive logic programs[10] and EHPP [27], and NDHPP naturally subsumes normal disjunctive logic programs [2] and NHPP [29].

Another reason why the proposed languages are interesting is that, in addition to allowing representing and reasoning with both qualitative and quantitative uncertainty, it can be also used in some real-world applications in which quantitative uncertainly need to be defined over qualitative uncertainty, where probabilistic measures are assigned over the possible outcomes of qualitative uncertainty. For example, flipping a fair coin leads to a head or tail with 0.5 probability each. This fact can be implicitly represented as a disjunctive logic program (since both events are equally likely) as $head(coin)$ *or* $tail(coin)$ with $\{head(coin)\}$ and $\{tail(coin)\}$ as the possible answer sets, according to the answer set semantics [10]. However, the explicit representation of probabilities and the explicit assignment of probabilities to the possible outcome of flipping the coin cannot be presented by disjunctive logic programs syntax and semantics. Moreover, consider the coin is biased to the head, where flipping the coin outcomes a head with 0.58 probability or a tail with 0.42 probability, in this case disjunctive logic program cannot represent it neither implicitly nor explicitly. On the other hand, the coin-flipping example cannot be represented intuitively and directly in NHPP or EHPP either, since a corresponding notion of disjunctions is not allowed in NHPP or EHPP.

2 Syntax

In this section we introduce the basic notions associated to the languages of EDHPP and NDHPP described throughout the rest of the paper [4,29,27]. EDHPP (NDHPP) are EHPP (NHPP) with disjunctions of annotated literals (atoms) in the head of rules.

Let $C[0,1]$ denotes the set of all closed intervals in $[0,1]$. In the context of EDHPP, probabilities are assigned to primitive events (literals) and compound events (conjunctions or disjunctions of literals) as intervals in $C[0,1]$. Let $[\alpha_1,\beta_1],[\alpha_2,\beta_2] \in C[0,1]$. Then the *truth order* asserts that $[\alpha_1,\beta_1] \leq_t [\alpha_2,\beta_2]$ iff $\alpha_1 \leq \alpha_2$ and $\beta_1 \leq \beta_2$. The set $C[0,1]$ and the relation $\leq_t$ form a complete lattice. In particular, the join ($\oplus_t$) operation is defined as $[\alpha_1,\beta_1] \oplus_t [\alpha_2,\beta_2] = [max\{\alpha_1,\alpha_2\}, max\{\beta_1,\beta_2\}]$ and the meet ($\otimes_t$) is defined as $[\alpha_1,\beta_1] \otimes_t [\alpha_2,\beta_2] = [min\{\alpha_1,\alpha_2\}, min\{\beta_1,\beta_2\}]$ w.r.t. $\leq_t$. The type of dependency among the primitive events within a compound event is described by *probabilistic strategies*, which are explicitly selected by the user. We call ρ, a pair of functions $\langle c, md\rangle$, a probabilistic strategy (p-strategy), where $c : C[0,1] \times C[0,1] \rightarrow C[0,1]$, the *probabilistic composition function*, which is *commutative*, *associative*, *monotonic* w.r.t. $\leq_t$, and meets the following *separation* criteria: there are two functions c_1, c_2 such that $c([\alpha_1,\beta_1],[\alpha_2,\beta_2]) = [c_1(\alpha_1,\alpha_2), c_2(\beta_1,\beta_2)]$. Whereas, $md : C[0,1] \rightarrow C[0,1]$ is the *maximal interval function*. The maximal interval function md of a certain p-strategy returns an estimate of the probability range of a primitive event, e, from the probability range of a compound event that contains e. The composition function c returns the probability range of a conjunction (disjunction) of two events given the ranges of its constituents. For convenience, given a multiset of probability intervals $M = \{\!\{[\alpha_1,\beta_1],\ldots,[\alpha_n,\beta_n]\}\!\}$,

we use cM to denote $c([\alpha_1, \beta_1], c([\alpha_2, \beta_2], \ldots, c([\alpha_{n-1}, \beta_{n-1}], [\alpha_n, \beta_n])) \ldots)$. According to the type of combination among events, p-strategies are classified into *conjunctive* p-strategies and *disjunctive* p-strategies. Conjunctive (disjunctive) p-strategies are employed to compose events belonging to a conjunctive (disjunctive) formula (please see [4,28] for the formal definitions).

Let $\mathcal{L}$ be an arbitrary first-order language with finitely many predicate symbols, function symbols, constants, and infinitely many variables. In addition, let $S = S_{conj} \cup S_{disj}$ be an arbitrary set of p-strategies, where S_{conj} (S_{disj}) is the set of all conjunctive (disjunctive) p-strategies in S. The Herbrand base of $\mathcal{L}$ is denoted by $\mathcal{B}_{\mathcal{L}}$. A literal is either an atom a or the negation of an atom $\neg a$, where $\neg$ is the classical negation. We denote the set of all literals in $\mathcal{L}$ by Lit. More formally, $Lit = \{a | a \in \mathcal{B}_{\mathcal{L}}\} \cup \{\neg a | a \in \mathcal{B}_{\mathcal{L}}\}$. An *annotation* denotes a probability interval and it is represented by $[\alpha_1, \alpha_2]$, where α_1, α_2 are called annotation items. An *annotation item* is either a constant in $[0, 1]$, a variable (*annotation variable*) ranging over $[0, 1]$, or $f(\alpha_1, \ldots, \alpha_n)$ (called *annotation function*) where f is a representation of a monotonic total function $f : ([0, 1])^n \rightarrow [0, 1]$ and $\alpha_1, \ldots, \alpha_n$ are annotation items.

The building blocks of the language of EDHPP are *hybrid literals*. Let us consider a set of literals $l_1, \ldots, l_n$ and the p-strategies ρ and ρ'. Then $l_1 \wedge_\rho \ldots \wedge_\rho l_n$ and $l_1 \vee_{\rho'} \ldots \vee_{\rho'} l_n$ are called *hybrid literals*. $bf_S(Lit)$ is the set of all ground hybrid literals formed using distinct literals from Lit and p-strategies from S, such that for any collection of equivalent hybrid literals, $\mathsf{Y} = \{l_1 *_\rho l_2 *_\rho \ldots *_\rho l_n,\ l_2 *_\rho l_1 *_\rho \ldots *_\rho l_n, \ldots\}$, where $* \in \{\wedge, \vee\}$, only one $l_{i_1} *_\rho l_{i_2} *_\rho \ldots *_\rho l_{i_n} \in \mathsf{Y}$ is in $bf_S(Lit)$. An *annotated hybrid literal* is an expression of the form $L : \mu$, where L is a hybrid literal and μ is an annotation. Note that any hybrid literal L can be represented in terms of another hybrid literal L' such that $L = \neg L'$, since $\neg\neg a = a$, $(a_1 \wedge_\rho a_2) = \neg(\neg a_1 \vee_\rho \neg a_2)$ and $(a_1 \vee_{\rho'} a_2) = \neg(\neg a_1 \wedge_{\rho'} \neg a_2)$ and $\wedge_\rho, \vee_\rho, \vee_{\rho'}$, and $\wedge_{\rho'}$ are associative and commutative.

The building blocks of the language of NDHPP are *hybrid basic formulae*. Let us consider a collection of atoms $a_1, \ldots, a_n$ and the p-strategies ρ and ρ'. Then $a_1 \wedge_\rho \ldots \wedge_\rho a_n$ and $a_1 \vee_{\rho'} \ldots \vee_{\rho'} a_n$ are called *hybrid basic formulae*. $bf_S(\mathcal{B}_{\mathcal{L}})$ is the set of all ground hybrid basic formulae formed using distinct atoms from $\mathcal{B}_{\mathcal{L}}$ and p-strategies from S, such that for any collection of equivalent hybrid basic formulae, $\mathsf{X} = \{a_1 *_\rho a_2 *_\rho \ldots *_\rho a_n,\ a_2 *_\rho a_1 *_\rho \ldots *_\rho a_n, \ldots\}$, where $* \in \{\wedge, \vee\}$, only one $a_{i_1} *_\rho a_{i_2} *_\rho \ldots *_\rho a_{i_n} \in \mathsf{X}$ is in $bf_S(\mathcal{B}_{\mathcal{L}})$. An annotated hybrid basic formula is an expression of the form $F : \mu$ where F is a hybrid basic formula and μ is an annotation.

3 Extended and Normal Disjunctive Hybrid Probabilistic Logic Programs

In this section we define the syntax, declarative semantics, the probabilistic answer sets semantics of *Extended Disjunctive Hybrid Probabilistic Logic Programs (EDHPP)*, and the stable probabilistic model semantics of *Normal Disjunctive Hybrid Probabilistic Logic Programs (NDHPP)*.

Definition 1 (Rules). *An extended disjunctive hybrid probabilistic rule (ed-rule) is an expression of the form*
$l_1 : \nu_1$ *or* $\ldots$ *or* $l_k : \nu_k \leftarrow L_1 : \mu_1, \ldots, L_m : \mu_m, not\ (L_{m+1} : \mu_{m+1}), \ldots, not\ (L_n : \mu_n)$,
whereas a normal disjunctive hybrid probabilistic rule (nd-rule) is an expression of the form
$A_1 : \nu_1$ *or* $\ldots$ *or* $A_k : \nu_k \leftarrow F_1 : \mu_1, \ldots, F_m : \mu_m, not\ (F_{m+1} : \mu_{m+1}), \ldots, not\ (F_n : \mu_n)$, *where* $l_1, \ldots, l_k$ *are literals,* $A_1, \ldots, A_k$ *are atoms,* L_i $(1 \leq i \leq n)$ *are hybrid literals,* F_i $(1 \leq i \leq n)$ *are hybrid basic formulae, and* $\nu_i (1 \leq i \leq k), \mu_i$ $(1 \leq i \leq n)$ *are annotations.*

An ed-rule^{not} *is an ed-rule without non-monotonic negation—i.e.,* $n = m$*, and a d-rule is an nd-rule without non-monotonic negation—i.e.,* $n = m$.

The intuitive meaning of an ed-rule, in Definition 1, is that, if for each $L_i : \mu_i$, where $1 \leq i \leq m$, the probability interval of L_i is at least μ_i and for each *not* $(L_j : \mu_j)$, where $m+1 \leq j \leq n$, it is *not known (undecidable)* that the probability interval of L_j is at least μ_j, then there exist at least l_i, where $1 \leq i \leq k$, such that the probability interval of l_i is at least ν_i. However, the meaning of an nd-rule, is that, if for each $F_i : \mu_i$, where $1 \leq i \leq m$, the probability interval of F_i is at least μ_i and for each *not* $(F_j : \mu_j)$, where $m + 1 \leq j \leq n$, it is *not provable* that the probability interval of F_j is at least μ_j, then there exist at least A_i, where $1 \leq i \leq k$, such that the probability interval of A_i is at least ν_i.

Definition 2 (Programs). *An extended (normal) disjunctive hybrid probabilistic logic program over* S, ed-program *(*nd-program*), is a pair* $P = \langle R, \tau \rangle$*, where* R *is a finite set of ed-rules (nd-rules) with p-strategies from* S*, and* τ *is a mapping* $\tau : Lit \rightarrow S_{disj}$ $(\tau : \mathcal{B}_{\mathcal{L}} \rightarrow S_{disj})$*. An extended (normal) disjunctive hybrid probabilistic logic program without non-monotonic negation is an ed-program (nd-program) where each rule in the program is an ed-rule*^{not} *(d-rule).*

The mapping τ in the above definition associates to each literal l_i (similarly for atoms in nd-programs) a disjunctive p-strategy that will be employed to combine the probability intervals obtained from different rules having l_i in their heads. An ed-program (nd-program) is ground if no variables appear in any of its rules.

3.1 Satisfaction and Models

In this subsection, we define the declarative semantics of EDHPP and NDHPP. We define the notions of interpretations, models, and satisfaction of ed-programs and nd-programs.

Definition 3. *A probabilistic interpretation (p-interpretation) of an ed-program is a partial or total mapping* $h : bf_S(Lit) \rightarrow C[0,1]$*. A probabilistic interpretation (p-inter pretation) for an nd-program is a total mapping* $h : bf_S(\mathcal{B}_{\mathcal{L}}) \rightarrow C[0,1]$.

Since we allow both an event and its negation to be defined in p-interpretations for ed-programs, more conditions need to be imposed on p-interpretations to ensure their consistency. This can be characterized by the following definitions.

Definition 4. *A total (partial) p-interpretation h for an ed-program is inconsistent if there exists $L, \neg L \in bf_S(Lit)$ $(L, \neg L \in dom(h))$ such that $h(\neg L) \neq [1,1] - h(L)$.*

Definition 5. *We say a set C, a subset of Lit, is a set of consistent literals if there is no pair of complementary literals a and $\neg a$ belonging to C. Similarly, a consistent set of hybrid literals C^* is a subset of $bf_S(Lit)$ such that there is no pair of complementary hybrid literals F and $\neg F$ belonging to C^*.*

Definition 6. *A consistent p-interpretation h of an ed-program is either not inconsistent or maps a consistent set of hybrid literals C^* to $C[0,1]$.*

The notion of truth order can be extended to p-interpretations of nd-programs. Given p-interpretations h_1 and h_2 of an nd-program P, we say $(h_1 \leq_t h_2) \Leftrightarrow (\forall F \in bf_S(\mathcal{B}_\mathcal{L}) : h_1(F) \leq_t h_2(F))$. The set of all p-interpretations of P and the truth order $\leq_t$ form a complete lattice. In addition, given the p-interpretations h_1 and h_2 for an ed-program P', we say $(h_1 \leq_o h_2) \Leftrightarrow (dom(h_1) \subseteq dom(h_2)$ *and* $\forall L \in dom(h_1),\ h_1(L) \leq_t h_2(L))$. The set of all p-interpretations of P' and the partial order $\leq_o$ form a complete lattice.

Definition 7 (Probabilistic Satisfaction). *Let $P = \langle R, \tau \rangle$ be a ground ed-program, h be a p-interpretation, and*

$r \equiv l_1 : \nu_1 or \ldots or\ l_k : \nu_k \leftarrow L_1 : \mu_1, \ldots, L_m : \mu_m, not\,(L_{m+1} : \mu_{m+1}), \ldots, not\ (L_n : \mu_n).$

Then

- *h satisfies $L_i : \mu_i (l_i : \nu_i)$ (denoted by $h \models L_i : \mu_i (h \models l_i : \nu_i)$) iff $L_i \in dom(h) (l_j \in dom(h))$ and $\mu_i \leq_t h(L_i) (\nu_i \leq_t h(l_i))$.*
- *h satisfies not $(L_j : \mu_j)$ (denoted by $h \models not\ (L_j : \mu_j)$) iff $L_j \in dom(h)$ and $\mu_j \not\leq_t h(L_j)$ or $L_j \notin dom(h)$.*
- *h satisfies $Body \equiv L_1 : \mu_1, \ldots, L_m : \mu_m, not\ (L_{m+1} : \mu_{m+1}), \ldots, not\ (L_n : \mu_n)$ (denoted by $h \models Body$) iff $\forall (1 \leq i \leq m), h \models L_i : \mu_i$ and $\forall (m+1 \leq j \leq n), h \models not\ (L_j : \mu_j)$.*
- *h satisfies $Head \equiv l_1 : \nu_1\ or \ldots or\ l_k : \nu_k$ (denoted by $h \models Head$) iff there exists at least i $(1 \leq i \leq k)$ such that $h \models l_i : \nu_i$.*
- *h satisfies $Head \leftarrow Body$ iff $h \models Head$ whenever $h \models Body$ or h does not satisfy $Body$.*
- *h satisfies P iff h satisfies every ed-rule in R and*
  - *If $l_i \in dom(h)$, $1 \leq i \leq k$, is a literal, then we have $c_{\tau(l_i)}\{\!\{\nu_i \mid l_1 : \nu_1\ or \ldots or\ l_k : \nu_k \leftarrow Body \in R, h \models Body,$ and $h \models l_i : \nu_i\}\!\} \leq_t h(l_i)$.*
  - *If $l_1, \ldots, l_n \in dom(h)$ are literals, then we have $L = l_1 *_\rho \ldots *_\rho l_n \in dom(h)$ and $c_\rho\{\!\{h(l_1), \ldots, h(l_n)\}\!\} \leq_t h(L)$.*

Observe that the definition of probabilistic satisfaction for nd-programs is the same as the definition of probabilistic satisfaction for ed-programs described in Definition 7. The only difference is that classical negation is not allowed in nd-programs, in addition, p-interpretations of nd-programs are total mappings from $bf_S(\mathcal{B}_\mathcal{L})$ to $C[0,1]$.

Definition 8 (Models). *A probabilistic model (*p-model*) of an ed-program (nd-program), with or without non-monotonic negation, P is a p-interpretation of P that satisfies P.*

Definition 9 (Minimal Models). *Let P be an ed-program (nd-program). A p-model h of P is* minimal *w.r.t. $\leq_o$ ($\leq_t$) iff there does not exist a p-model h' of P such that $h' <_o h$ ($h' <_t h$).*

We call a minimal p-model of an ed-program a *probabilistic answer set*. It is possible to get a probabilistic answer set of an ed-program, P, and this probabilistic answer set is inconsistent. If this is the case, we say P is inconsistent. If P is inconsistent, LIT, where $LIT : bf_S(Lit) \rightarrow [1,1]$, is the probabilistic answer set of P. We adopt this view from the answer sets semantics of classical logic programming [10].

Example 1. Consider the following ed-program $P = \langle R, \tau \rangle$, without non-monotonic negation, where R contains

$a : [0.1, 0.2] \quad or \ \neg b : [0.15, 0.3] \quad \neg c : [0, 0.21] \quad \leftarrow a : [0.1, 0.13]$
$d : [0.12, 0.18] \leftarrow \neg b : [0.1, 0.21] \quad \neg d : [0.45, 0.55] \leftarrow a : [0, 0.15], \neg b : [0.02, 0.22], \neg c : [0.1, 0.1]$

and τ is any arbitrary assignment of disjunctive p-strategies. It is easy to verify that P has two probabilistic answer sets h_1 and h_2, where
$h_1(a) = [0.1, 0.2] \quad h_1(\neg c) = [0, 0.21]$ and $h_2(\neg b) = [0.15, 0.3] \quad h_2(d) = [0.12, 0.18]$.

3.2 Probabilistic Answer Sets and Stable Probabilistic Model Semantics

In this subsection we define the *probabilistic answer set* and the *stable probabilistic model* semantics of ed-programs and nd-programs respectively. The semantics are defined in two steps. First, we guess a probabilistic answer set (stable probabilistic model) h for a certain ed-program (nd-program) P, then we define the notion of the probabilistic reduct of P with respect to h. The probabilistic reduct is an ed-program (nd-program) without non-monotonic negation. Second, we determine whether h is a probabilistic answer set (stable probabilistic model) for P. This is verified by determining whether h is a probabilistic answer set (minimal p-model) of the probabilistic reduct of P w.r.t. h.

Definition 10 (Probabilistic Reduct). *Let $P = \langle R, \tau \rangle$ be a ground ed-program (nd-program) and h be a p-interpretation. The probabilistic reduct P^h of P w.r.t. h is $P^h = \langle R^h, \tau \rangle$ where:*

$$R^h = \left\{ l_1 : \nu_1 \ or \ldots or \ l_k : \nu_k \leftarrow L_1 : \mu_1, \ldots, L_m : \mu_m \;\middle|\; \begin{array}{l} l_1 : \nu_1 \ or \ldots or \ l_k : \nu_k \leftarrow L_1 : \mu_1, \ldots, L_m : \mu_m, \\ not \ (L_{m+1} : \mu_{m+1}), \ldots, not \ (L_n : \mu_n) \in R \ and \\ \forall (m+1 \leq j \leq n), \ \mu_j \not\leq_t h(L_j) \ or \ L_j \notin dom(h) \end{array} \right\}$$

Note that the definitions of the probabilistic reduct for ed-programs and nd-programs are similar. Except that classical negation is not allowed in nd-programs.

In addition, p-interpretations in nd-programs are total mappings from $bf_S(\mathcal{B}_\mathcal{L})$ to $C[0,1]$, therefore, for nd-programs, the condition $L_j \notin dom(h)$ is not applicable.

The probabilistic reduct P^h is an ed-program (nd-program) without non-monotonic negation. For any *not* $(L_j : \mu_j)$ in the body of $r \in R$ with $\mu_j \not\leq_t h(L_j)$ means that it is *not known (not provable for nd-program)* that the probability interval of L_j is at least μ_j given the available knowledge, and *not* $(L_j : \mu_j)$ is removed from the body of r. In addition, for ed-program, if $L_j \notin dom(h)$, i.e., L_j is undefined in h, then it is completely *not known (undecidable)* that the probability interval of L_j is at least μ_j. In this case, *not* $(L_j : \mu_j)$ is also removed from the body of r. If $\mu_j \leq_t h(L_j)$ (similarly for nd-programs), then we know that the probability interval of L_j is at least μ_j and the body of r is not satisfied and r is trivially ignored.

Definition 11. *A p-interpretation h of an ed-program (nd-program) P is a probabilistic answer set (stable probabilistic model) of P if h is a minimal p-model of P^h.*

The domain of a probabilistic answer set of an ed-program or a stable probabilistic model of an nd-program represents an agent set of beliefs. However, the probability intervals associated to these beliefs bound the agents belief degrees on these beliefs.

ed-programs without classical negation (nd-programs), i.e., ed-programs that contain no negative literals neither in head nor in the body of ed-rules, have probabilistic answer sets with hybrid literals consist of only atoms (hybrid basic formulae). Moreover, the definition of probabilistic answer sets coincides with the definition of stable probabilistic models for nd-programs. This means that the application of the probabilistic answer sets semantics to nd-programs is reduced to the stable probabilistic model semantics for nd-programs. However, there are a couple of main differences between the two semantics. A probabilistic answer set may be a partial p-interpretation, however, a stable probabilistic model is a total p-interpretation. In addition, each hybrid basic formula F with probability interval [0,0] in a stable probabilistic model of an nd-program corresponds to the fact that the probability interval of F is unknown, and hence undefined, in its equivalent probabilistic answer set.

Proposition 1. *Let P be an ed-program without classical negation. Then h is a probabilistic answer set for P iff h' is a stable probabilistic model of P, where $h(F) = h'(F)$ for each $h'(F) \neq [0,0]$ and $h(F)$ is undefined for each $h'(F) = [0,0]$.*

Proposition 1 suggests that there is a simple reduction from ed-programs to nd-programs. The importance of that is, under the consistency condition, computational methods developed for nd-programs can be applied to ed-programs.

Example 2. Consider the following example adapted from [11]. Tom and Fred are two policemen who are challenging their firing gun skills, by shooting a bottle at a quite long distance. In one of the shoots, at the same time, both Tom and Fred shoot a bottle and the bottle shattered. In fact, we cannot determine whether

Tom or Fred is the one who shattered the bottle. However, from Tom's shooting experience on similar targets at similar distances, Tom is capable of hitting targets with probability interval from 75% to 80%. Similarly, Fred can hit similar targets with probability interval from 72% to 87%. Normally, a shooter shoots a target. If a shooter sneezes while shooting, it is an exception. Hence, a shooter's shoot is abnormal with probability interval from 30% to 65% if a shooter sneezes while shooting. It was heard that somebody sneezed, however, we do not know whether Tom or Fred is the one who sneezed. A shooter shatters a bottle with probability interval from 82% to 90% if a shooter is capable of hitting similar targets with probability interval from 70% to 79%, and it is not known that a shooter's shoot is abnormal with probability interval from 30% to 60%. This can be represented by the following ed-program $P = \langle R, \tau \rangle$, where R contains:

$$
\begin{array}{lcl}
sneeze(tom) : [1,1] & or & sneeze(fred) : [1,1] \leftarrow \\
ab(shoot, X) : [0.3, 0.65] & \leftarrow & shoot(X) : [1,1], sneeze(X) : [1,1] \\
shatter(X) : [0.82, 0.9] & \leftarrow & hit(X) : [0.7, 0.79], not\ (ab(shoot, X) : [0.3, 0.6]) \\
shoot(tom) : [1,1] \leftarrow & & shoot(fred) : [1,1] \leftarrow \\
hit(tom) : [0.75, 0.8] \leftarrow & & hit(fred) : [0.72, 0.87] \leftarrow
\end{array}
$$

and τ is any arbitrary assignment of disjunctive p-strategies. The ed-rules in Example 2 encode two forms of uncertainty. Qualitative uncertainty represented by the first ed-rule that arises from the fact that we do not know whether Tom or Fred is the one who sneezed. And quantitative uncertainty represented by the probability intervals associated to the various events presented in R. The probability interval $[1,1]$ represents the truth value *true*. Therefore, the rule $sneeze(tom) : [1,1]$ *or* $sneeze(fred) : [1,1] \leftarrow$ is intuitively interpreted as a disjunctive rule in classical disjunctive logic programming. The above ed-program P has two probabilistic answer sets h_1 and h_2, where

$$
\begin{array}{lll}
h_1(sneeze(fred)) & = [1,1] & h_2(sneeze(tom)) = [1,1] \\
h_1(ab(shoot, fred)) & = [0.3,0.65] & h_2(ab(shoot, tom)) = [0.3,0.65] \\
h_1(shatter(tom)) & = [0.82,\ 0.9] & h_2(shatter(fred)) = [0.82,\ 0.9] \\
h_1(shoot(tom)) & = [1,1] & h_2(shoot(tom)) = [1,1] \\
h_1(shoot(fred)) & = [1,1] & h_2(shoot(fred)) = [1,1] \\
h_1(hit(tom)) & = [0.75,0.8] & h_2(hit(fred)) = [0.72,0.87] \\
h_1(hit(fred)) & = [0.72,0.87] & h_2(hit(tom)) = [0.75,0.8]
\end{array}
$$

For example, h_1 can be verified as a probabilistic answer set of P by computing the probabilistic reduct, $P^{h_1} = \langle R^{h_1}, \tau \rangle$, of P w.r.t. h_1, where R^{h_1} contains

$$
\begin{array}{lcl}
sneeze(tom) : [1,1] & or & sneeze(fred) : [1,1] \leftarrow \\
ab(shoot, tom) : [0.3, 0.65] & \leftarrow & shoot(tom) : [1,1], sneeze(tom) : [1,1] \\
ab(shoot, fred) : [0.3, 0.65] & \leftarrow & shoot(fred) : [1,1], sneeze(fred) : [1,1] \\
shatter(tom) : [0.82, 0.9] & \leftarrow & hit(tom) : [0.7, 0.79] \\
shoot(tom) : [1,1] \leftarrow & & shoot(fred) : [1,1] \leftarrow \\
hit(tom) : [0.75, 0.8] \leftarrow & & hit(fred) : [0.72, 0.87] \leftarrow
\end{array}
$$

It can be easily seen that h_1 is a probabilistic answer set for P^{h_1}. Now we show that EDHPP and NDHPP naturally extend EHPP and NHPP respectively.

Proposition 2. *The probabilistic answer sets semantics of EDHPP is equivalent to the probabilistic answer sets semantics of EHPP [27] for all ed-programs* $P = \langle R, \tau \rangle$ *such that* $\forall\, r \in R, k = 1$. *In addition, the stable probabilistic model semantics of NDHPP is equivalent to the stable probabilistic model semantics of NHPP [29] for all nd-programs* $P = \langle R, \tau \rangle$ *such that* $\forall\, r \in R, k = 1$.

Let us show that the probabilistic answer sets semantics of EDHPP and the stable probabilistic model semantics of NDHPP generalize the answer sets semantics and the stable model semantics of extended and normal disjunctive logic programs [2,10] respectively. An extended disjunctive logic program P can be represented as an ed-program $P' = \langle R, \tau \rangle$ where each extended disjunctive rule

$$l_1 \; or \ldots or \; l_k \leftarrow l'_1, \ldots, l'_m, not \; l'_{m+1}, \ldots, not \; l'_n \in P$$

can be represented, in R, as an ed-rule of the form

$$l_1 : [1,1] \; or \ldots or \; l_k : [1,1] \leftarrow l'_1 : [1,1], \ldots, l'_m : [1,1], not \; (l'_{m+1} : [1,1]), \ldots, not \; (l'_n : [1,1]) \in R$$

where $l_1, \ldots, l_k, l'_1, \ldots, l'_n$ are literals and $[1,1]$ represents the truth value *true*. τ is any arbitrary assignment of disjunctive p-strategies. We call the class of ed-programs that consists of only ed-rules of the above form as $EDHPP_1$. Recall that nd-programs are ed-programs without classical negation. $NDHPP_1$ is the same as $EDHPP_1$, except that, only atoms (positive literals) are allowed to appear in rules of the above form. The following result shows that $EDHPP_1$ and $NDHPP_1$ subsume classical extended and normal disjunctive logic programs [2,10].

Proposition 3. *Let* P_1 *be an extended disjunctive logic program. Then* ${S_1}'$ *is an answer set of* P_1 *iff* h_1 *is a probabilistic answer of* ${P_1}' \in EDHPP_1$ *that corresponds to* P_1 *where* $h_1(l) = [1,1]$ *iff* $l \in {S_1}'$ *and* $h_1(l')$ *is undefined iff* $l' \notin {S_1}'$. *Let* P_2 *be a normal disjunctive logic program. Then* ${S_2}'$ *is a stable model of* P_2 *iff* h_2 *is a stable probabilistic model of* ${P_2}' \in NDHPP_1$ *that corresponds to* P_2 *where* $h_2(a) = [1,1]$ *iff* $a \in {S_2}'$ *and* $h_2(b) = [0,0]$ *iff* $b \in \mathcal{B}_{\mathcal{L}} \setminus {S_2}'$.

In [1], a logical approach has been presented to reason with causal Bayes networks, by considering a body of logical knowledge, using the answer sets semantics of classical logic programming [1]. Answer sets semantics has been used to emulate the possible world semantics in [1]. In the following result, we show that EDHPP naturally subsumes the probabilistic logic programming framework (P-log) of [1]. This means that any P-log program can be represented as an ed-program.

Proposition 4. *The language of EDHPP subsumes P-log, a probabilistic logic programming framework for reasoning with causal Bayes networks [1].*

4 Related Work

Many approaches have been proposed to augment logic programming with qualitative or quantitative uncertainty. Qualitative uncertainty is presented in logicprogramming by disjunctive logic programs [19,10,2], where EDHPP (NDHPP) subsumes. However, quantitative uncertainty is represented in logic programming by various formalisms including probability theory. Although, representing and reasoning with both forms of uncertainty is needed [6], this issue has not been addressed by the current work in qualitative or quantitative uncertainty in logic programming. The main difference in this work is that we allow representing and reasoning in the presence of both qualitative (represented by disjunctions) and quantitative uncertainty (represented by probability theory) in a unified logic programming framework. In addition, we allow the assignment of quantitative uncertainly over qualitative uncertainty, where probabilistic measures are assigned over the possible outcomes of qualitative uncertainty. The current work in the literature supports either qualitative uncertainty [19,10,2] or quantitative uncertainty [12,20,25,26,30,21,22,23,4,17,18,3,28,29,27].

The closest to our work are the frameworks presented in [30,18,1].

Although [30] allows disjunctions in the head of rules, the probabilistic logic programming framework in [30] is used to represent and reason with quantitative uncertainty to reason with Bayes networks. In addition, EDHPP (NDHPP) is more expressive than [30], since, for example EDHPP, unlike [30], allows classical negation, non-monotonic negation, different modes of probabilistic combinations (since [30] considers independence of probabilities which is a fixed mode of probabilistic combination), and compound events to appear in the body of rules, as well as, Bayes reasoning and representation.

Similar to [30], another approach for probabilistic logic programming has been provided in [18] for quantitative uncertainty reasoning. In [18], a possible world semantics for reasoning about probabilities has been introduced by assigning probabilistic measures over the possible worlds using normal disjunctive logic programs. A probabilistic logic program in [18] consists of a set of normal disjunctive logic program clauses with associated probabilities. A normal disjunctive clause in [18] is treated as a classical formula with an associated probability, where the implication in such a clause is treated as material implication. In addition, an approximate semantics for probabilistic logic programming in [18] has been presented, where probabilities are treated as a lattice of truth values. In this case, the probability of a conjunction $Prob(A \wedge B) = min(Prob(A), Prob(B))$ and the probability of a disjunction $Prob(A \vee B) = max(Prob(A), Prob(B))$. This is considered a fixed mode of combination. Whereas, in our framework conjunctions and disjunctions are treated differently according to the type of dependency between events. In addition, unlike [18], we allow classical negation and compound events to appear in the body of rules.

A logical approach has been presented in [1] to reason with causal Bayes networks by considering a body of logical knowledge, by using the answer sets semantics of classical answer set programming [10]. Although, full answer set programming (logic programs with classical negation, non-monotonic negation,

and disjunctions) is used, the probabilistic logic programming framework in [1] is used to reason in the presence of quantitative uncertainty. Answer sets semantics [10] has been used to emulate the possible world semantics. Probabilistic logic programs of [1] is expressive and straightforward and relaxed some restrictions on the logical knowledge representation part existed in similar approaches to Bayesian reasoning, e.g., [12,20,25,26,30]. Since [21,22,23,4] provided a different semantical characterization to probabilistic logic programming, it was not clear that how these proposals relate to [1]. However, the work presented in this paper and [29,27], which are modification and generalization of the work presented in [21,22,23,4], are closely related to [1]. The framework presented in this paper, as well as the framework of [27], is strictly syntactically and semantically subsumes probabilistic logic programs of [1]. This can be argued by the fact that EDHPP naturally extends classical extended disjunctive logic programs with answer sets semantics [10], and probabilistic logic programs of [1] mainly rely on extended disjunctive logic programs with answer sets semantics [10] as a knowledge representation and inference mechanism for reasoning with causal Bayes networks. In this sense, the comparisons established between [1] and the existing probabilistic logic programming approaches such as [12,20,25,26,30,21,22,23,4,17,18,3] also carry over to EDHPP and these approaches. In addition, unlike [1], EDHPP does not put any restriction on the type of dependency existing among events.

5 Conclusions and Future Work

We extended Extended and Normal Hybrid Probabilistic Logic Programs [27,29] to Extended and Normal Disjunctive Hybrid Probabilistic Logic Programs, to allow classical negation, non-monotonic negation, and disjunctions in the head of rules. The extension is necessary to provide the capability of reasoning in the presence of both qualitative and quantitative uncertainty in a unified logic programming framework. In addition to the ability to assign quantitative uncertainly over qualitative uncertainty, where probabilistic measures are assigned over the possible outcomes of qualitative uncertainty. We developed semantical characterizations of the extended languages, which rely on generalizations of the answer sets semantics and the stable model semantics, originally developed for extended and normal disjunctive logic programs [10,2], and the probabilistic answer sets semantics and the stable probabilistic model semantics for Extended and Normal Hybrid Probabilistic Logic Programs [27,29]. We showed that the probabilistic answer sets semantics of EDHPP naturally generalizes the answer sets semantics of extended disjunctive logic programs [10] and the probabilistic answer sets semantics of EHPP [27]. In addition, the stable probabilistic model semantics of NDHPP generalizes the stable model semantics of normal disjunctive logic programs [2] and the stable probabilistic model semantics of NHPP [29]. Furthermore, we showed that the probabilistic answer sets semantics of EDHPP is reduced to stable probabilistic model semantics of NDHPP. The importance of that is computational methods developed for NDHPP can be applied to the

language of EDHPP. Moreover, we showed that some commonsense probabilistic knowledge can be easily represented in EDHPP and NDHPP. In addition, we showed that EDHPP naturally subsumes the probabilistic logicprogramming framework of [1]. The main topic of future research is to develop algorithms and implementations for computing the semantics of EDHPP and NDHPP.

References

1. Baral, C., et al.: Probabilistic reasoning with answer sets. In: LPNMR (2004)
2. Brewka, G., Dix, J.: Knowledge representation with logic programs. In: Dix, J., Moniz Pereira, L., Przymusinski, T.C. (eds.) LPKR 1997. LNCS (LNAI), vol. 1471, Springer, Heidelberg (1998)
3. Dekhtyar, A., Dekhtyar, I.: Possible worlds semantics for probabilistic logic programs. In: Demoen, B., Lifschitz, V. (eds.) ICLP 2004. LNCS, vol. 3132, pp. 137–148. Springer, Heidelberg (2004)
4. Dekhtyar, A., Subrahmanian, V.S.: Hybrid probabilistic program. Journal of LP 43(3), 187–250 (2000)
5. Dekhtyar, M., Dekhtyar, A., Subrahmanian, V.S.: Hybrid Probabilistic Programs: Algorithms and Complexity. In: UAI, pp. 160–169 (1999)
6. Eiter, T., Lukasiewicz, T.: Probabilistic reasoning about actions in nonmonotonic causal theories. In: UAI, pp. 192–199 (2003)
7. Eiter, T., et al.: Declarative problem solving in dlv. In: LBAI (2000)
8. Van Gelder, A., Ross, K.A., Schlipf, J.S.: The well-founded semantics for general logic programs. Journal of ACM 38(3), 620–650 (1991)
9. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: ICSLP, MIT Press, Cambridge (1988)
10. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Computing 9(3-4), 363–385 (1991)
11. Halpern, J., Pearl, J.: Causes and explanations: A structural-model approach. Part I: Causes. The British Journal for the Philosophy of Science 56(4), 843–887 (2005)
12. Kersting, K., De Raedt, L.: Bayesian Logic Programs. In: Inductive LP (2000)
13. Kifer, M., Subrahmanian, V.S.: Theory of generalized annotated logic programming and its applications. Journal of Logic Programming 12, 335–367 (1992)
14. Lakshmanan, L.V.S., Sadri, F.: On a theory of probabilistic deductive databases. Journal of TPLP 1(1), 5–42 (2001)
15. Loyer, Y., Straccia, U.: The well-founded semantics in normal logic programs with uncertainty. In: Hu, Z., Rodríguez-Artalejo, M. (eds.) FLOPS 2002. LNCS, vol. 2441, Springer, Heidelberg (2002)
16. Loyer, Y., Straccia, U.: The approximate well-founded semantics for logic programs with uncertainty. In: Int. Symp. on Math. Found. of CS (2003)
17. Lukasiewicz, T.: Probabilistic logic programming. In: JELIA, pp. 388–392 (1998)
18. Lukasiewicz, T.: Many-valued disjunctive logic programs with probabilistic semantics. In: Gelfond, M., Leone, N., Pfeifer, G. (eds.) LPNMR 1999. LNCS (LNAI), vol. 1730, Springer, Heidelberg (1999)
19. Fernandez, J., Minker, J.: Disjunctive deductive databases. In: Voronkov, A. (ed.) LPAR 1992. LNCS, vol. 624, Springer, Heidelberg (1992)
20. Muggleton, S.: Stochastic logic programming. In: International Workshop on ILP (1995)
21. Ng, R.T., Subrahmanian, V.S.: Probabilistic logic programming. Information & Computation 101(2) (1992)

22. Ng, R.T., Subrahmanian, V.S.: A semantical framework for supporting subjective and conditional probabilities in deductive databases. ARJ 10(2) (1993)
23. Ng, R.T., Subrahmanian, V.S.: Stable semantics for probabilistic deductive databases. Information & Computation 110(1) (1994)
24. Niemela, I., Simons, P.: Efficient implementation of the well-founded and stable model semantics. In: Joint International Conference and Symposium on LP (1996)
25. Poole, D.: The Independent choice logic for modelling multiple agents under uncertainty. Artificial Intelligence 94(1-2), 7–56 (1997)
26. Poole, D.: Abducing through negation as failure: stable models within the independent choice logic. Journal of Logic Programming 44, 5–35 (2000)
27. Saad, E.: Incomplete knowlege in hybrid probabilistic logic programs. In: JELIA (2006)
28. Saad, E., Pontelli, E.: Towards a more practical hybrid probabilistic logic programming framework. In: PADL (2005)
29. Saad, E., Pontelli, E.: A new approach to hybrid probabilistic logic programs. Annals of Mathematics and Artificial Intelligence Journal 48(3-4), 187–243 (2006)
30. Vennekens, J., Verbaeten, S., Bruynooghe, M.: Logic programs with annotated disjunctions. In: Demoen, B., Lifschitz, V. (eds.) ICLP 2004. LNCS, vol. 3132, Springer, Heidelberg (2004)

Description Logic Programs Under Probabilistic Uncertainty and Fuzzy Vagueness

Thomas Lukasiewicz[1,2] and Umberto Straccia[3]

[1] DIS, Sapienza Università di Roma, Via Ariosto 25, I-00185 Rome, Italy
lukasiewicz@dis.uniroma1.it
[2] Institut für Informationssysteme, TU Wien, Favoritenstraße 9-11, A-1040 Vienna, Austria
lukasiewicz@kr.tuwien.ac.at
[3] ISTI-CNR, Via G. Moruzzi 1, I-56124 Pisa, Italy
straccia@isti.cnr.it

Abstract. This paper is directed towards an infrastructure for handling both uncertainty and vagueness in the Rules, Logic, and Proof layers of the Semantic Web. More concretely, we present probabilistic fuzzy description logic programs, which combine fuzzy description logics, fuzzy logic programs (with stratified nonmonotonic negation), and probabilistic uncertainty in a uniform framework for the Semantic Web. We define important concepts dealing with both probabilistic uncertainty and fuzzy vagueness, such as the expected truth value of a crisp sentence and the probability of a vague sentence. Furthermore, we describe a shopping agent example, which gives evidence of the usefulness of probabilistic fuzzy description logic programs in realistic web applications. In the extended report, we also provide algorithms for query processing in probabilistic fuzzy description logic programs, and we delineate a special case where query processing can be done in polynomial time in the data complexity.

1 Introduction

The *Semantic Web* [1,4] aims at an extension of the current World Wide Web by standards and technologies that help machines to understand the information on the Web so that they can support richer discovery, data integration, navigation, and automation of tasks. The main ideas behind it are to add a machine-readable meaning to web pages, to use ontologies for a precise definition of shared terms in web resources, to use KR technology for automated reasoning from web resources, and to apply cooperative agent technology for processing the information of the Web.

The Semantic Web consists of several hierarchical layers, where the *Ontology layer*, in form of the *OWL Web Ontology Language* [19], is currently the highest layer of sufficient maturity. OWL consists of three increasingly expressive sublanguages, namely *OWL Lite*, *OWL DL*, and *OWL Full*. OWL Lite and OWL DL are essentially very expressive description logics with an RDF syntax. As shown in [8], ontology entailment in OWL Lite (resp., OWL DL) reduces to knowledge base (un)satisfiability in the description logic $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$). On top of the Ontology layer, sophisticated representation and reasoning capabilities for the *Rules*, *Logic*, and *Proof layers* of the Semantic Web are being developed next.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 187–198, 2007.

In particular, a key requirement of the layered architecture of the Semantic Web is to integrate the Rules and the Ontology layer. Here, it is crucial to allow for building rules on top of ontologies, that is, for rule-based systems that use vocabulary from ontology knowledge bases. Another type of combination is to build ontologies on top of rules, where ontological definitions are supplemented by rules or imported from rules. Both types of integration have been realized in recent hybrid integrations of rules and ontologies under the loose coupling, called *(loosely coupled) description logic programs* (or simply *dl-programs*), which have the form $KB = (L, P)$, where L is a description logic knowledge base, and P is a finite set of rules involving queries to L [3].

Other research efforts are directed towards *handling uncertainty and vagueness in the Semantic Web*, which are motivated by important web and semantic web applications. In particular, formalisms for handling uncertainty are used in data integration, ontology mapping, and information retrieval, while dealing with vagueness is motivated by multimedia information processing / retrieval and natural language interfaces to the Web. There are several extensions of description logics and web ontology languages by probabilistic uncertainty and fuzzy vagueness. Similarly, there are also extensions of description logic programs by probabilistic uncertainty [9] and fuzzy vagueness [16,10].

Clearly, since uncertainty and vagueness are semantically quite different, it is important to have a unifying formalism for the Semantic Web, which allows for dealing with both uncertainty and vagueness. But even though there has been some important work in the fuzzy logic community in this direction [5], to date there are no approaches to description logic programs that allow for handling both uncertainty and vagueness.

In this paper, we try to fill this gap. We present a novel approach to description logic programs, where probabilistic rules are defined on top of fuzzy rules, which are in turn defined on top of fuzzy description logics. This approach allows for handling both probabilistic uncertainty and fuzzy vagueness. Intuitively, it allows for defining several rankings on ground atoms using fuzzy vagueness, and then for merging these rankings using probabilistic uncertainty (by associating with each ranking a probabilistic weight and building the weighted sum of all rankings). The main contributions are as follows:

- We present probabilistic fuzzy description logic programs, which combine (i) fuzzy description logics, (ii) fuzzy logic programs (with stratified default negation), and (iii) probabilistic uncertainty in a uniform framework for the Semantic Web.
- Such programs allow for handling both probabilistic uncertainty (especially for probabilistic ontology mapping and probabilistic data integration) and fuzzy vagueness (especially for dealing with vague concepts). We define important concepts dealing with both probabilistic uncertainty and fuzzy vagueness, such as the expected truth value of a crisp sentence and the probability of a vague sentence.
- We describe a shopping agent example, which gives evidence of the usefulness of probabilistic fuzzy description logic programs in realistic web applications.
- In the extended report [11], we also give algorithms for query processing in probabilistic fuzzy description logic programs, and we delineate a special case where query processing is data tractable (under suitable assumptions about the underlying fuzzy description logics), which is an important feature for the Web.

The rest of this paper is organized as follows. Section 2 gives a motivating example. In Section 3, we recall combination strategies and fuzzy description logics. Section 4

defines fuzzy dl-programs on top of fuzzy description logics. In Section 5, we then define probabilistic fuzzy dl-programs. Section 6 summarizes our main results and gives an outlook on future research. Note that algorithms for query processing and data tractability results as well as further technical details are given in the extended report [11].

2 Motivating Example

In this section, we describe a shopping agent example, where we encounter both probabilistic uncertainty (in resource selection, ontology mapping / query transformation, and data integration) and fuzzy vagueness (in query matching with vague concepts).

Example 2.1 (Shopping Agent). Suppose a person would like to buy "a sports car that costs at most about 22 000 € and that has a power of around 150 HP".

In todays Web, the buyer has to *manually* (i) search for car selling sites, e.g., using Google, (ii) select the most promising sites (e.g., http://www.autos.com), (iii) browse through them, query them to see the cars that they sell, and match the cars with our requirements, (iv) select the offers in each web site that match our requirements, and (v) eventually merge all the best offers from each site and select the best ones.

It is obvious that the whole process is rather *tedious* and *time consuming*, since e.g. (i) the buyer has to visit many sites, (ii) the browsing in each site is very time consuming, (iii) finding the right information in a site (which has to match the requirements) is not simple, and (iv) the way of browsing and querying may differ from site to site.

A *shopping agent* may now support us as follows, *automatizing* the whole selection process once it receives the request / query q from the buyer:

- *Probabilistic Resource Selection.* The agent selects some sites / resources S that it considers as promising for the buyer's request. The agent has to select a subset of some *relevant* resources, since it is not reasonable to assume that it will access and query all the resources known to him. The relevance of a resource S to a query is usually (automatically) estimated as the probability $Pr(q|S)$ (the probability that the information need represented by the query q is satisfied by the searching resource S, see e.g. [2,6]). It is not difficult to see that such probabilities can be represented by probabilistic rules.
- *Probabilistic Ontology Mapping / Query Reformulation.* For the top-k selected sites, the agent has to reformulate the buyer's query using the terminology / ontology of the specific car selling site. For this task, the agent relies on so-called transformation rules, which say how to translate a concept or property of the agent's ontology into the ontology of the information resource (which is called *ontology mapping* in the Semantic Web). To relate a concept B of the buyer's ontology to a concept S of the seller's ontology, one often automatically estimates the probability $P(B|S)$ that an instance of S is also an instance of B, which can then be represented as a probabilistic rule [17,18,12].
- *Vague Query Matching.* Once the agent has translated the buyer's request for the specific site's terminology, the agent submits the query. But the buyer's request often contains many so-called *vague / fuzzy* concepts such as "the price is around 22 000 € or less", rather than strict conditions, and thus a car may *match* the buyer's

Table 1. Combination strategies of various fuzzy logics

	Łukasiewicz Logic	Gödel Logic	Product Logic	Zadeh Logic
$a \otimes b$	$\max(a+b-1,0)$	$\min(a,b)$	$a \cdot b$	$\min(a,b)$
$a \oplus b$	$\min(a+b,1)$	$\max(a,b)$	$a+b-a \cdot b$	$\max(a,b)$
$a \rhd b$	$\min(1-a+b,1)$	$\begin{cases} 1 & \text{if } a \leqslant b \\ b & \text{otherwise} \end{cases}$	$\min(1,b/a)$	$\max(1-a,b)$
$\ominus a$	$1-a$	$\begin{cases} 1 & \text{if } a=0 \\ 0 & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } a=0 \\ 0 & \text{otherwise} \end{cases}$	$1-a$

condition to a *degree*. As a consequence, a site / resource / web service may return a ranked list of cars, where the ranks depend on the degrees to which the sold items match the buyer's requests q.

– *Probabilistic Data Integration.* Eventually, the agent has to combine the ranked lists (see e.g. [14]) by considering the involved matching (or truth) degrees (vagueness) and probability degrees (uncertainty) and show the top-n items to the buyer.

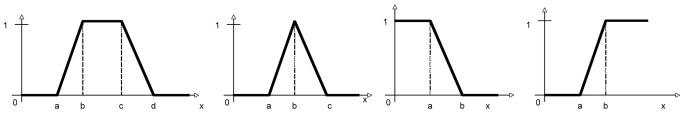
3 Preliminaries

In this section, we review combination strategies and fuzzy description logics, mainly through some examples; more details are given in the extended report [11].

Combination Strategies. Rather than being restricted to a binary truth value among **false** and **true**, *vague propositions* may also have a truth value strictly between **false** and **true**. In the sequel, we use the unit interval $[0,1]$ as the set of all possible truth values, where 0 and 1 represent the ordinary binary truth values **false** and **true**, respectively. For example, the vague proposition "John is a tall man" may be more or less true, and it is thus associated with a truth value in $[0,1]$, depending on the body height of John. To combine and modify the truth values in $[0,1]$, we assume *combination strategies*, namely, *conjunction*, *disjunction*, *implication*, and *negation strategies*, denoted $\otimes$, $\oplus$, $\rhd$, and $\ominus$, respectively, which are functions $\otimes, \oplus, \rhd\colon [0,1] \times [0,1] \to [0,1]$ and $\ominus\colon [0,1] \to [0,1]$ that generalize the ordinary Boolean operators $\wedge$, $\vee$, $\rightarrow$, and $\neg$, respectively, to the set of truth values $[0,1]$. As usual, we assume that combination strategies have some natural algebraic properties. Note that conjunction and disjunction strategies are also called *triangular norms* and *triangular co-norms* [7], respectively.

Example 3.1. The combination strategies of various fuzzy logics are shown in Table 1.

Fuzzy Description Logics. Intuitively, description logics model a domain of interest in terms of concepts and roles, which represent classes of individuals resp. binary relations between classes of individuals. A knowledge base encodes in particular subset relationships between concepts, subset relationships between roles, the membership of individuals to concepts, and the membership of pairs of individuals to roles. In fuzzy description logics, these relationships and memberships then have a degree of truth in $[0,1]$.



(a) (b) (c) (d)

Fig. 1. (a) Tra-function, (b) Tri-function, (c) L-function, and (d) R-function

We assume fuzzy generalizations of the description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$ behind OWL Lite and OWL DL, respectively. We now describe the syntax of fuzzy $\mathcal{SHIF}(\mathbf{D})$ and fuzzy $\mathcal{SHOIN}(\mathbf{D})$ (see especially [15]) and illustrate it through an example. For a formal semantics and more details see [11]; for an implementation of fuzzy $\mathcal{SHIF}(\mathbf{D})$, the *fuzzyDL* system, see http://gaia.isti.cnr.it/∼straccia.

The elementary ingredients are as follows. We assume a set of *data values*, a set of *elementary datatypes*, and a set of *datatype predicates* (each with a predefined arity $n \geqslant 1$). A *datatype* is an elementary datatype or a finite set of data values. A *fuzzy datatype theory* $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$ consists of a datatype domain $\Delta^{\mathbf{D}}$ and a mapping $\cdot^{\mathbf{D}}$ that assigns to each data value an element of $\Delta^{\mathbf{D}}$, to each elementary datatype a subset of $\Delta^{\mathbf{D}}$, and to each datatype predicate of arity n a fuzzy relation over $\Delta^{\mathbf{D}}$ of arity n (that is, a mapping $(\Delta^{\mathbf{D}})^n \rightarrow [0,1]$). We extend $\cdot^{\mathbf{D}}$ to all datatypes by $\{v_1, \ldots, v_n\}^{\mathbf{D}} = \{v_1^{\mathbf{D}}, \ldots, v_n^{\mathbf{D}}\}$. Non-crisp predicates are usually defined by functions for specifying fuzzy set membership degrees, such as the trapezoidal, triangular, left shoulder, and right shoulder functions (see Fig. 1). Let $\mathbf{A}$, $\mathbf{R}_A$, $\mathbf{R}_D$, $\mathbf{I}$, and $\mathbf{M}$ be pairwise disjoint sets of *atomic concepts*, *abstract roles*, *datatype roles*, *individuals*, and *fuzzy modifiers*, respectively.

A *role* is any element of $\mathbf{R}_A \cup \mathbf{R}_A^- \cup \mathbf{R}_D$ (where $\mathbf{R}_A^-$ is the set of *inverses* R^- of all $R \in \mathbf{R}_A$). We define *concepts* inductively as follows. Each $A \in \mathbf{A}$ is a concept, $\bot$ and $\top$ are concepts, and if $a_1, \ldots, a_n \in \mathbf{I}$, then $\{a_1, \ldots, a_n\}$ is a concept (called *oneOf*). If C, C_1, C_2 are concepts, $R, S \in \mathbf{R}_A \cup \mathbf{R}_A^-$, and $m \in \mathbf{M}$, then $(C_1 \sqcap C_2)$, $(C_1 \sqcup C_2)$, $\neg C$, and $m(C)$ are concepts (called *conjunction*, *disjunction*, *negation*, and *fuzzy modification*, respectively), as well as $\exists R.C$, $\forall R.C$, $\geqslant nS$, and $\leqslant nS$ (called *exists*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geqslant 0$. If D is a datatype and $T, T_1, \ldots, T_n \in \mathbf{R}_D$, then $\exists T_1, \ldots, T_n.D$, $\forall T_1, \ldots, T_n.D$, $\geqslant nT$, and $\leqslant nT$ are concepts (called *datatype exists*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geqslant 0$. We eliminate parentheses as usual.

A *crisp axiom* has one of the following forms: (1) $C \sqsubseteq D$ (called *concept inclusion axiom*), where C and D are concepts; (2) $R \sqsubseteq S$ (called *role inclusion axiom*), where either $R, S \in \mathbf{R}_A \cup \mathbf{R}_A^-$ or $R, S \in \mathbf{R}_D$; (3) $\mathrm{Trans}(R)$ (called *transitivity axiom*), where $R \in \mathbf{R}_A$; (4) $C(a)$ (called *concept assertion axiom*), where C is a concept and $a \in \mathbf{I}$; (5) $R(a,b)$ (resp., $U(a,v)$) (called *role assertion axiom*), where $R \in \mathbf{R}_A$ (resp., $U \in \mathbf{R}_D$) and $a, b \in \mathbf{I}$ (resp., $a \in \mathbf{I}$ and v is a data value); and (6) $a = b$ (resp., $a \neq b$) (*equality* (resp., *inequality*) *axiom*), where $a, b \in \mathbf{I}$. We define *fuzzy axioms* as follows: A *fuzzy concept inclusion* (resp., *fuzzy role inclusion*, *fuzzy concept assertion*, *fuzzy role assertion*) *axiom* is of the form $\alpha \, \theta \, n$, where α is a concept inclusion (resp., role inclusion, concept assertion, role assertion) axiom, $\theta \in \{\leqslant, =, \geqslant\}$, and $n \in [0,1]$. Informally, $\alpha \leqslant n$ (resp., $\alpha = n$, $\alpha \geqslant n$) encodes that the truth value of α is at most (resp., equal to,

at least) n. We often use α to abbreviate $\alpha = 1$. A *fuzzy (description logic) knowledge base* L is a finite set of fuzzy axioms, transitivity axioms, and equality and inequality axioms. For decidability, number restrictions in L are restricted to simple abstract roles. Notice that L may contain fuzzy concept inclusion axioms (between general concepts).

Fuzzy $\mathcal{SHIF}(\mathbf{D})$ has the same syntax as fuzzy $\mathcal{SHOIN}(\mathbf{D})$, but without the oneOf constructor and with the atleast and atmost constructors limited to 0 and 1.

Example 3.2 (Shopping Agent cont'd). The following axioms are an excerpt of the fuzzy description logic knowledge base L that conceptualizes the site in Example 2.1:

$$\mathit{Cars} \sqcup \mathit{Trucks} \sqcup \mathit{Vans} \sqcup \mathit{SUVs} \sqsubseteq \mathit{Vehicles}\,, \tag{1}$$

$$\mathit{PassengerCars} \sqcup \mathit{LuxuryCars} \sqsubseteq \mathit{Cars}\,, \tag{2}$$

$$\mathit{CompactCars} \sqcup \mathit{MidSizeCars} \sqcup \mathit{SportyCars} \sqsubseteq \mathit{PassengerCars}\,, \tag{3}$$

$$\begin{aligned}\mathit{Cars} \sqsubseteq\ & (\exists \mathit{hasReview}.\mathbf{Integer}) \sqcap\ (\exists \mathit{hasInvoice}.\mathbf{Integer}) \sqcap\ (\exists \mathit{hasHP}.\mathbf{Integer}) \\ & \sqcap\ (\exists \mathit{hasResellValue}.\mathbf{Integer}) \sqcap\ (\exists \mathit{hasSafetyFeatures}.\mathbf{Integer}) \sqcap\ \ldots\,,\end{aligned} \tag{4}$$

$$(\mathit{SportyCar} \sqcap (\exists \mathit{hasInvoice}.\{18883\}) \sqcap (\exists \mathit{hasHP}.\{166\}) \sqcap \ldots)(\mathit{MazdaMX5Miata})\,, \tag{5}$$

$$(\mathit{SportyCar} \sqcap (\exists \mathit{hasInvoice}.\{20341\}) \sqcap (\exists \mathit{hasHP}.\{200\}) \sqcap \ldots)(\mathit{VolkswagenGTI})\,, \tag{6}$$

$$(\mathit{SportyCar} \sqcap (\exists \mathit{hasInvoice}.\{24029\}) \sqcap (\exists \mathit{hasHP}.\{162\}) \sqcap \ldots)(\mathit{MitsubishiES})\,. \tag{7}$$

Here, axioms (1)–(3) describe the concept taxonomy of the site, while axiom (4) describes the datatype attributes of the cars sold in the site. For example, every passenger or luxury car is also a car, and every car has a resell value. Axioms (5)–(7) describe the properties of some sold cars. For example, the $\mathit{MazdaMX5Miata}$ is a sports car, costing 18 883 €. Note that $\mathbf{Integer}$ denotes the datatype of all integers.

We may now encode "costs at most about 22 000 €" and "has a power of around 150 HP" in the buyer's request through the following concepts C and D, respectively:

$$C = \exists \mathit{hasInvoice}.\mathit{LeqAbout22000} \ \text{ and } \ D = \exists \mathit{hasHP}.\mathit{Around150HP}\,,$$

where $\mathit{LeqAbout22000} = L(22000, 25000)$ and $\mathit{Around150HP} = \mathit{Tri}(125, 150, 175)$ (see Fig. 1). The latter two equations define the fuzzy concepts "at most about 22 000 €" resp. "around 150 HP". The former is modeled as a left shoulder function stating that if the price is less than 22 000, then the degree of truth (degree of buyer's satisfaction) is 1, else the truth is linearly decreasing to 0 (reached at the cost of 25 000). In fact, we are modeling a case were the buyer would like to pay less than 22 000, though may still accept a higher price (up to 25 000) to a lesser degree. Similarly, the latter models the fuzzy concept "around 150 HP" as a triangular function with vertice in 150 HP.

The following fuzzy axioms are (tight) logical consequences of the above description logic knowledge base L (under the Zadeh semantics of the connectives):

$$\begin{aligned}&C(\mathit{MazdaMX5Miata}) = 1.0\,,\ C(\mathit{VolkswagenGTI}) = 1.0\,,\ C(\mathit{MitsubishiES}) = 0.32\,,\\ &D(\mathit{MazdaMX5Miata}) = 0.36\,,\ D(\mathit{VolkswagenGTI}) = 0.0\,,\ D(\mathit{MitsubishiES}) = 0.56\,.\end{aligned}$$

4 Fuzzy Description Logic Programs

In this section, we define fuzzy dl-programs, which are similar to the fuzzy dl-programs in [10], except that they are based on fuzzy description logics as in [15], and that we

consider only stratified fuzzy dl-programs here. Their canonical model associates with every ground atom a truth value, and so defines a ranking on the Herbrand base. We first introduce the syntax, and we then define the semantics of positive and stratified fuzzy dl-programs in terms of a least model resp. an iterative least model semantics.

Syntax. Informally, a normal fuzzy program is a finite collection of normal fuzzy rules, which are similar to ordinary normal rules, except that (i) they have a lower bound for their truth value, and (ii) they refer to fuzzy interpretations rather than binary interpretations, and thus every of their logical operators (that is, "$\leftarrow$", "$\wedge$", and "not") is associated with a combination strategy (that is, "$\leftarrow$" and "$\wedge$" are associated with a conjunction strategy $\otimes$, while "not" is associated with a negation strategy $\ominus$) to specify how the operator combines truth values. Formally, we assume a function-free first-order vocabulary Φ with finite nonempty sets of constant symbols (which also belong to the set $\mathbf{I}$ of all description logic individuals) and predicate symbols, and a set $\mathcal{X}$ of variables. A *term* is a constant symbol from Φ or a variable from $\mathcal{X}$. If p is a predicate symbol of arity $k \geqslant 0$ from Φ, and $t_1, \ldots, t_k$ are terms, then $p(t_1, \ldots, t_k)$ is an *atom*. A *literal* is an atom a or a default-negated atom $not\, a$. A *normal fuzzy rule* r has the form

$$\begin{aligned} a \leftarrow_{\otimes_0} b_1 \wedge_{\otimes_1} b_2 \wedge_{\otimes_2} \cdots \wedge_{\otimes_{k-1}} b_k \wedge_{\otimes_k} \\ not_{\ominus_{k+1}}\, b_{k+1} \wedge_{\otimes_{k+1}} \cdots \wedge_{\otimes_{m-1}} not_{\ominus_m}\, b_m \geqslant v\,, \end{aligned} \tag{8}$$

where $m \geqslant k \geqslant 0$, $a, b_1, \ldots, b_m$ are atoms, $\otimes_0, \ldots, \otimes_{m-1}$ are conjunction strategies, $\ominus_{k+1}, \ldots, \ominus_m$ are negation strategies, and $v \in [0, 1]$. We call a the *head* of r, denoted $H(r)$, while the conjunction $b_1 \wedge_{\otimes_1} \ldots \wedge_{\otimes_{m-1}} not_{\ominus_m}\, b_m$ is the *body* of r. We define $B(r) = B^+(r) \cup B^-(r)$, where $B^+(r) = \{b_1, \ldots, b_k\}$ and $B^-(r) = \{b_{k+1}, \ldots, b_m\}$. A *normal fuzzy program* P is a finite set of normal fuzzy rules.

Informally, a fuzzy dl-program consists of a fuzzy description logic knowledge base L and a generalized normal fuzzy program P, which may contain queries to L. In such a query, it is asked whether a concept or a role assertion logically follows from L or not (see [3] for more background and examples of such queries). Formally, a *dl-query* $Q(\mathbf{t})$ is either (a) of the form $C(t)$, where C is a concept, and t is a term, or (b) of the form $R(t_1, t_2)$, where R is a role, and t_1 and t_2 are terms. A *dl-atom* has the form $DL[S_1 \uplus p_1, \ldots, S_m \uplus p_m; Q](\mathbf{t})$, where each S_i is an atomic concept or a role, p_i is a unary resp. binary predicate symbol, $Q(\mathbf{t})$ is a dl-query, and $m \geqslant 0$. We call $p_1, \ldots, p_m$ its *input predicate symbols*. Intuitively, $S_i \uplus p_i$ encodes that the truth value of every $S_i(\mathbf{e})$ is at least the truth value of $p_i(\mathbf{e})$, where $\mathbf{e}$ is a constant (resp., pair of constants) from Φ when S_i is a concept (resp., role) (and thus p_i is a unary (resp., binary) predicate symbol). A *fuzzy dl-rule* r is of the form (8), where any b_i in the body of r may be a dl-atom. A *fuzzy dl-program* $KB = (L, P)$ consists of a satisfiable fuzzy description logic knowledge base L and a finite set of fuzzy dl-rules P. *Substitutions*, *ground substitutions*, *ground terms*, *ground atoms*, etc., are defined as usual. We denote by $ground(P)$ the set of all ground instances of fuzzy dl-rules in P relative to Φ.

Example 4.1 (Shopping Agent cont'd). A fuzzy dl-program $KB = (L, P)$ is given by the fuzzy description logic knowledge base L in Example 3.2, and the set of fuzzy dl-rules P, which contains only the following fuzzy dl-rule encoding the buyer's request, where $\otimes$ may, e.g., be the Gödel conjunction strategy (that is, $x \otimes y = \min(x, y)$):

$$query(x) \leftarrow_{\otimes} SportyCar(x) \wedge_{\otimes} hasInvoice(x, y_1) \wedge_{\otimes} hasHP(x, y_2) \wedge_{\otimes} DL[LeqAbout22000](y_1) \wedge_{\otimes} DL[Around150HP](y_2) \geqslant 1\,.$$

Models of Fuzzy DL-Programs. We first define fuzzy (Herbrand) interpretations, the semantics of dl-queries, and the truth of fuzzy dl-rules and of fuzzy dl-programs in interpretations. In the sequel, let $KB = (L, P)$ be a (fully general) fuzzy dl-program.

We denote by HB_Φ (resp., HU_Φ) the Herbrand base (resp., universe) over Φ. In the sequel, we assume that HB_Φ is nonempty. A *fuzzy interpretation* I is a mapping $I\colon HB_\Phi \to [0,1]$. We denote by $\boldsymbol{HB}_\Phi$ the fuzzy interpretation I such that $I(a) = 1$ for all $a \in HB_\Phi$. For fuzzy interpretations I and J, we write $I \subseteq J$ iff $I(a) \leqslant J(a)$ for all $a \in HB_\Phi$, and we define the *intersection* of I and J, denoted $I \cap J$, by $(I \cap J)(a) = \min(I(a), J(a))$ for all $a \in HB_\Phi$. Note that $I \subseteq \boldsymbol{HB}_\Phi$ for all fuzzy interpretations I. The truth value of $a \in HB_\Phi$ in I under L, denoted $I_L(a)$, is defined as $I(a)$. The truth value of a ground dl-atom $a = DL[S_1 \uplus p_1, \ldots, S_m \uplus p_m; Q](\mathbf{c})$ in I under L, denoted $I_L(a)$, is the supremum of v subject to $L \cup \bigcup_{i=1}^{m} A_i(I) \models Q(\mathbf{c}) \geqslant v$ and $v \in [0,1]$, where $A_i(I) = \{S_i(\mathbf{e}) \geqslant I(p_i(\mathbf{e})) \mid I(p_i(\mathbf{e})) > 0,\ p_i(\mathbf{e}) \in HB_\Phi\}$. We say I is a *model* of a ground fuzzy dl-rule r of form (8) under L, denoted $I \models_L r$, iff

$$I_L(a) \geqslant I_L(b_1) \otimes_1 I_L(b_2) \otimes_2 \cdots \otimes_{k-1} I_L(b_k) \otimes_k \ominus_{k+1} I_L(b_{k+1}) \otimes_{k+1} \cdots \otimes_{m-1} \ominus_m I_L(b_m) \otimes_0 v\,,$$

Here, we implicitly assume that $\otimes_1, \ldots, \otimes_{m-1}, \otimes_0$ are evaluated from left to right. We say I is a *model* of $KB = (L, P)$, denoted $I \models KB$, iff $I \models_L r$ for all $r \in ground(P)$.

Positive Fuzzy DL-Programs. Informally, positive fuzzy dl-programs have no default negation: A fuzzy dl-program $KB = (L, P)$ is *positive* iff P is "not"-free.

For ordinary positive programs, as well as positive dl-programs KB, the intersection of a set of models of KB is also a model of KB. A similar result holds for positive fuzzy dl-programs KB. Hence, every positive fuzzy dl-program KB has as its *canonical model* a unique least model, denoted M_{KB}, which is contained in every model of KB.

Example 4.2 (Shopping Agent cont'd). The fuzzy dl-program $KB = (L, P)$ of Example 4.1 is positive, and its minimal model M_{KB} is given as follows:

$$M_{KB}(query(MazdaMX5Miata)) = 0.36\,, \quad M_{KB}(query(MitsubishiES)) = 0.32\,,$$

and all other ground instances of $query(x)$ have the truth value 0 under M_{KB}.

Stratified Fuzzy DL-Programs. We next define stratified fuzzy dl-programs, which are composed of hierarchic layers of positive fuzzy dl-programs that are linked via default negation. Like for ordinary stratified programs, as well as stratified dl-programs, a minimal model can be defined by a finite number of iterative least models, which naturally describes as the *canonical model* the semantics of stratified fuzzy dl-programs.

For any fuzzy dl-program $KB = (L, P)$, let DL_P denote the set of all ground dl-atoms that occur in $ground(P)$. An *input atom* of $a \in DL_P$ is a ground atom with an input predicate of a and constant symbols in Φ. A *stratification of* $KB = (L, P)$ (*with respect to* DL_P) is a mapping $\lambda\colon HB_\Phi \cup DL_P \to \{0, 1, \ldots, k\}$ such that

(i) $\lambda(H(r)) \geqslant \lambda(a)$ (resp., $\lambda(H(r)) > \lambda(a)$) for each $r \in \mathit{ground}(P)$ and $a \in B^+(r)$ (resp., $a \in B^-(r)$), and
(ii) $\lambda(a) \geqslant \lambda(a')$ for each input atom a' of each $a \in DL_P$,

where $k \geqslant 0$ is the *length* of λ. For $i \in \{0, \ldots, k\}$, we define $KB_i = (L, P_i) = (L, \{r \in \mathit{ground}(P) \mid \lambda(H(r)) = i\})$, and we define HB_{P_i} (resp., $HB^\star_{P_i}$) as the set of all $a \in HB_\Phi$ such that $\lambda(a) = i$ (resp., $\lambda(a) \leqslant i$).

A fuzzy dl-program $KB = (L, P)$ is *stratified* iff it has a stratification λ of some length $k \geqslant 0$. We define its iterative least models $M_i \subseteq \boldsymbol{HB}_\Phi$ with $i \in \{0, \ldots, k\}$ by:

(i) M_0 is the least model of KB_0;
(ii) if $i > 0$, then M_i is the least model of KB_i such that $M_i|HB^\star_{P_{i-1}} = M_{i-1}|HB^\star_{P_{i-1}}$, where $M_i|HB^\star_{P_{i-1}}$ and $M_{i-1}|HB^\star_{P_{i-1}}$ denote the restrictions of the mappings M_i and M_{i-1} to $HB^\star_{P_{i-1}}$, respectively.

Then, M_{KB} denotes M_k. Note that M_{KB} is well-defined, since it does not depend on a particular stratification λ. Furthermore, M_{KB} is in fact a minimal model of KB.

5 Probabilistic Fuzzy Description Logic Programs

In this section, we introduce probabilistic fuzzy dl-programs as a combination of stratified fuzzy dl-programs with Poole's independent choice logic (ICL) [13]. This will allow us to express probabilistic rules. Poole's ICL is based on ordinary acyclic logic programs P under different "atomic choices", where each atomic choice along with P produces a first-order model, and one then obtains a probability distribution on the set of first-order models by placing a probability distribution on the different atomic choices. Here, we use stratified fuzzy dl-programs rather than ordinary acyclic logic programs, and thus we define a probability distribution on a set of fuzzy interpretations. In other words, we define a probability distribution on a set of rankings on the Herbrand base.

Syntax. We now define the syntax of probabilistic fuzzy dl-programs and probabilistic queries addressed to them. We first introduce fuzzy formulas, query constraints, and probabilistic formulas, and we define choice spaces and probabilities on choice spaces.

We define *fuzzy formulas* by induction as follows. The propositional constants *false* and *true*, denoted $\bot$ and $\top$, respectively, and all atoms $p(t_1, \ldots, t_k)$ are fuzzy formulas. If ϕ and ψ are fuzzy formulas, and $\otimes$, $\oplus$, $\rhd$, and $\ominus$ are conjunction, disjunction, implication, resp. negation strategies, then $(\phi \wedge_\otimes \psi)$, $(\phi \vee_\oplus \psi)$, $(\phi \Rightarrow_\rhd \psi)$, and $\neg_\ominus \phi$ are also fuzzy formulas. A *query constraint* has the form $(\phi\, \theta\, r)[l, u]$ or $(\mathbf{E}[\phi])[l, u]$ with $\theta \in \{\geqslant, >, <, \leqslant\}$, $r, l, u \in [0, 1]$, and fuzzy formulas ϕ. Informally, the former asks for the interval of the probability that the truth value v of ϕ satisfies $v\, \theta\, r$, while the latter asks for the interval of the expected truth value of ϕ. We define *probabilistic formulas* inductively as follows. Each query constraint is a probabilistic formula. If F and G are probabilistic formulas, then also $\neg F$ and $(F \wedge G)$. We use $(F \vee G)$ and $(F \Rightarrow G)$ to abbreviate $\neg(\neg F \wedge \neg G)$ resp. $\neg(F \wedge \neg G)$, and eliminate parentheses as usual.

A *choice space* C is a set of pairwise disjoint and nonempty sets $A \subseteq HB_\Phi$. Any $A \in C$ is an *alternative* of C and any $a \in A$ an *atomic choice* of C. Intuitively, every $A \in C$ represents a random variable and every $a \in A$ one of its possible values. A *total*

choice of C is a set $B \subseteq HB_\Phi$ such that $|B \cap A| = 1$ for all $A \in C$. Intuitively, every total choice B of C represents an assignment of values to all the random variables. A *probability* μ on a choice space C is a probability function on the set of all total choices of C. Intuitively, every μ is a probability distribution over the set of all variable assignments. Since C and all its alternatives are finite, μ can be defined by (i) a mapping $\mu\colon \bigcup C \to [0,1]$ such that $\sum_{a \in A} \mu(a) = 1$ for all $A \in C$, and (ii) $\mu(B) = \Pi_{b \in B} \mu(b)$ for all total choices B of C. Intuitively, (i) defines a probability over the values of each random variable of C, and (ii) assumes independence between the random variables.

A *probabilistic fuzzy dl-program* $KB = (L, P, C, \mu)$ consists of a stratified fuzzy dl-program (L, P), a choice space C such that (i) $\bigcup C \subseteq HB_\Phi$ and (ii) no atomic choice in C coincides with the head of any fuzzy dl-rule in $ground(P)$, and a probability μ on C. Intuitively, since the total choices of C select subsets of P, and μ is a probability distribution on the total choices of C, every probabilistic fuzzy dl-program compactly represents a probability distribution on a finite set of stratified fuzzy dl-programs. A *probabilistic query* to KB has the form $\exists F$, or $\exists(\alpha\, \theta\, r)[L, U]$, or $\exists(\mathbf{E}[\alpha])[L, U]$, where F is a probabilistic formula, α is a fuzzy formula, $r \in [0,1]$, and L, U are variables.

Example 5.1 (Shopping Agent cont'd). A probabilistic fuzzy dl-program $KB = (L, P, C, \mu)$ is given by L of Example 3.2, the following set of fuzzy dl-rules P, which model the query reformulation and retrieval steps using ontology mapping rules:

$$\begin{aligned} query(x) \leftarrow_\otimes\ & SportsCar(x) \wedge_\otimes hasPrice(x, y_1) \wedge_\otimes hasPower(x, y_2) \wedge_\otimes \\ & DL[LeqAbout22000](y_1) \wedge_\otimes DL[Around150HP](y_2) \geqslant 1\,, \qquad (9) \\ SportsCar(x) \leftarrow_\otimes\ & DL[SportyCar](x) \wedge_\otimes sc_{pos} \geqslant 0.9\,, \qquad (10) \\ hasPrice(x) \leftarrow_\otimes\ & DL[hasInvoice](x) \wedge_\otimes hi_{pos} \geqslant 0.8\,, \qquad (11) \\ hasPower(x) \leftarrow_\otimes\ & DL[hasHP](x) \wedge_\otimes hhp_{pos} \geqslant 0.8\,, \qquad (12) \end{aligned}$$

the choice space $C = \{\{sc_{pos}, sc_{neg}\}, \{hi_{pos}, hi_{neg}\}, \{hhp_{pos}, hhp_{neg}\}\}$, and the probability distribution μ, which is given by the following probabilities for the atomic choices sc_{pos}, sc_{neg}, hi_{pos}, hi_{neg}, hhp_{pos}, and hhp_{neg} (which are 0-ary predicate symbols), and then extended to all total choices by assuming independence:

$$\begin{aligned} &\mu(sc_{pos}) = 0.91\,, \quad \mu(sc_{neg}) = 0.09\,, \quad \mu(hi_{pos}) = 0.78\,, \\ &\mu(hi_{neg}) = 0.22\,, \quad \mu(hhp_{pos}) = 0.83\,, \quad \mu(hhp_{neg}) = 0.17\,. \end{aligned}$$

Intuitively, C encodes three probabilistically independent random variables with the binary domains $\{sc_{pos}, sc_{neg}\}$, $\{hi_{pos}, hi_{neg}\}$, and $\{hhp_{pos}, hhp_{neg}\}$. Rule (9) is the buyer's request, but in a "different" terminology than the one of the car selling site. Rules (10)–(12) are so-called ontology alignment mapping rules. For example, rule (10) states that the predicate "SportsCar" of the buyer's terminology refers to the concept "SportyCar" of the selected side, with probability 0.91. Such mapping rules can be automatically built by relying on ontology alignment tools, such as oMap [17,18], whose main purpose is to find relations among the concepts and roles of two different ontologies. oMap is particularly suited for our case, as it is based on a probabilistic model, and thus the mappings have a probabilistic reading (see also [12]).

Semantics. A *world* I is a fuzzy interpretation over HB_Φ. We denote by $\mathcal{I}_\Phi$ the set of all worlds over Φ. A *variable assignment* σ maps each $X \in \mathcal{X}$ to some $t \in HU_\Phi$.

It is extended to all terms by $\sigma(c) = c$ for all constant symbols c from Φ. The *truth value* of fuzzy formulas ϕ in I under σ, denoted $I_\sigma(\phi)$ (or $I(\phi)$ when ϕ is ground), is inductively defined by (1) $I_\sigma(\phi \wedge_\otimes \psi) = I_\sigma(\phi) \otimes I_\sigma(\psi)$, (2) $I_\sigma(\phi \vee_\oplus \psi) = I_\sigma(\phi) \oplus I_\sigma(\psi)$, (3) $I_\sigma(\phi \Rightarrow_\rhd \psi) = I_\sigma(\phi) \rhd I_\sigma(\psi)$, and (4) $I_\sigma(\neg_\ominus \phi) = \ominus I_\sigma(\phi)$.

A *probabilistic interpretation* Pr is a probability function on $\mathcal{I}_\Phi$ (that is, a mapping $Pr\colon \mathcal{I}_\Phi \to [0,1]$ such that (i) the set of all $I \in \mathcal{I}_\Phi$ with $Pr(I) > 0$ is denumerable, and (ii) all $Pr(I)$ with $I \in \mathcal{I}_\Phi$ sum up to 1). The *probability* of $\phi\, \theta\, r$ in Pr under a variable assignment σ, denoted $Pr_\sigma(\phi\, \theta\, r)$ (or $Pr(\phi\, \theta\, r)$ when ϕ is ground), is the sum of all $Pr(I)$ such that $I \in \mathcal{I}_\Phi$ and $I_\sigma(\phi)\, \theta\, r$. The *expected truth value* of ϕ under Pr and σ, denoted $\mathbf{E}_{Pr,\sigma}[\phi]$, is the sum of all $Pr(I) \cdot I_\sigma(\phi)$ with $I \in \mathcal{I}_\Phi$. Notice that in the notion of expected truth value, we combine probabilities and truth values. The *truth* of probabilistic formulas F in Pr under σ, denoted $Pr \models_\sigma F$, is inductively defined by (1) $Pr \models_\sigma (\phi\, \theta\, r)[l,u]$ iff $Pr_\sigma(\phi\, \theta\, r) \in [l,u]$, (2) $Pr \models_\sigma (\mathbf{E}[\phi])[l,u]$ iff $\mathbf{E}_{Pr,\sigma}[\phi] \in [l,u]$, (3) $Pr \models_\sigma \neg F$ iff not $Pr \models_\sigma F$, and (4) $Pr \models_\sigma (F \wedge G)$ iff $Pr \models_\sigma F$ and $Pr \models_\sigma G$.

A probabilistic interpretation Pr is a *model* of a probabilistic formula F iff $Pr \models_\sigma F$ for every variable assignment σ. We say Pr is the *canonical model* of a probabilistic fuzzy dl-program $KB = (L, P, C, \mu)$ iff every world $I \in \mathcal{I}_\Phi$ with $Pr(I) > 0$ is the canonical model of $(L, P \cup \{p \leftarrow \mid p \in B\})$ for some total choice B of C with $Pr(I) = \mu(B)$. Notice that every KB has a unique canonical model Pr. We say F is a *consequence* of KB, denoted $KB \mid\!\sim F$, iff the canonical model of KB is also a model of F. A query constraint $(\phi\, \theta\, r)[l,u]$ (resp., $(\mathbf{E}[\phi])[l,u]$) is a *tight consequence* of KB, denoted $KB \mid\!\sim_{tight} (\phi\, \theta r)[l,u]$ (resp., $KB \mid\!\sim_{tight} (\mathbf{E}[\phi])[l,u]$), iff l (resp., u) is the infimum (resp., supremum) of $Pr_\sigma(\phi\, \theta\, r)$ (resp., $\mathbf{E}_{Pr,\sigma}[\phi]$) subject to the canonical model Pr of KB and all σ. A *correct answer* to $\exists F$ is a substitution σ such that $F\sigma$ is a consequence of KB. A *tight answer* to $\exists(\alpha\, \theta\, r)[L,U]$ (resp., $\exists(\mathbf{E}[\alpha])[L,U]$) is a substitution σ such that $(\alpha\, \theta\, r)[L,U]\sigma$ (resp., $(\mathbf{E}[\alpha])[L,U]\sigma$) is a tight consequence of KB.

Example 5.2 (Shopping Agent cont'd). The following are some tight consequences of the probabilistic fuzzy dl-program $KB = (L, P, C, \mu)$ in Example 5.1:

$$(\mathbf{E}[query(MazdaMX5Miata)])[0.21, 0.21]\,,\ \ (\mathbf{E}[query(MitsubishiES)])[0.19, 0.19]\,.$$

So, the agent ranks the $MazdaMX5Miata$ first with degree 0.21 ($= 0.36 \cdot 0.91 \cdot 0.78 \cdot 0.83$) and the $MitsubishiES$ second with degree 0.19 ($= 0.32 \cdot 0.91 \cdot 0.78 \cdot 0.83$).

6 Summary and Outlook

We have presented probabilistic fuzzy dl-programs for the Semantic Web, which allow for handling both probabilistic uncertainty (especially for probabilistic ontology mapping and probabilistic data integration) and fuzzy vagueness (especially for dealing with vague concepts) in a uniform framework. We have defined important concepts related to both probabilistic uncertainty and fuzzy vagueness. Furthermore, we have described a shopping agent example, which gives evidence of the usefulness of probabilistic fuzzy dl-programs in realistic web applications. In the extended report [11], we also provide algorithms for query processing in such programs, which can be done in polynomial time in the data complexity under suitable assumptions.

An interesting topic of future research is to generalize probabilistic fuzzy dl-programs by non-stratified default negations, classical negations, and disjunctions. Another interesting issue is to explore how to update probabilistic fuzzy dl-programs.

Acknowledgments. This work has been partially supported by the German Research Foundation (DFG) under the Heisenberg Programme.

References

1. Berners-Lee, T.: Weaving the Web. Harper, San Francisco (1999)
2. Callan, J.: Distributed information retrieval. In: Croft, W.B. (ed.) Advances in Information Retrieval, pp. 127–150. Kluwer, Dordrecht (2000)
3. Eiter, T., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the Semantic Web. In: Proc. KR-2004, pp. 141–151 (2004)
4. Fensel, D., Wahlster, W., Lieberman, H., Hendler, J. (eds.): Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential. MIT Press, Cambridge (2002)
5. Flaminio, T., Godo, L.: A logic for reasoning about the probability of fuzzy events. Fuzzy Sets and Systems 158(6), 625–638 (2007)
6. Fuhr, N.: A decision-theoretic approach to database selection in networked IR. ACM Trans. Inf. Syst. 3(17), 229–249 (1999)
7. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
8. Horrocks, I., Patel-Schneider, P.F.: Reducing OWL entailment to description logic satisfiability. In: Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 17–29. Springer, Heidelberg (2003)
9. Lukasiewicz, T.: Probabilistic description logic programs. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 737–749. Springer, Heidelberg (2005) (Extended version in Int. J. Approx. Reason. 45(2):288–307, 2007)
10. Lukasiewicz, T.: Fuzzy description logic programs under the answer set semantics for the Semantic Web. In: Proc. RuleML-2006, pp. 89–96 (2006) (Extended version in Fundamenta Informaticae (to appear))
11. Lukasiewicz, T., Straccia, U.: Uncertainty and vagueness in description logic programs for the Semantic Web. Report 1843-07-02, Institut für Informationssysteme, TU Wien (2007)
12. Nottelmann, H., Straccia, U.: Information retrieval and machine learning for probabilistic schema matching. Inf. Process. Manage. 43(3), 552–576 (2007)
13. Poole, D.: The independent choice logic for modelling multiple agents under uncertainty. Artif. Intell. 94(1-2), 7–56 (1997)
14. Renda, M.E., Straccia, U.: Web metasearch: Rank vs. score-based rank aggregation methods. In: Proc. SAC-2003, pp. 841–846 (2003)
15. Straccia, U.: A fuzzy description logic for the Semantic Web. In: Sanchez, E. (ed.) Fuzzy Logic and the Semantic Web, Capturing Intelligence, ch. 4, pp. 73–90. Elsevier, Amsterdam (2006)
16. Straccia, U.: Fuzzy description logic programs. In: Proc. IPMU-2006, pp. 1818–1825 (2006)
17. Straccia, U., Troncy, R.: oMAP: Combining classifiers for aligning automatically OWL ontologies. In: Ngu, A.H.H., Kitsuregawa, M., Neuhold, E.J., Chung, J.-Y., Sheng, Q.Z. (eds.) WISE 2005. LNCS, vol. 3806, pp. 133–147. Springer, Heidelberg (2005)
18. Straccia, U., Troncy, R.: Towards distributed information retrieval in the Semantic Web. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 378–392. Springer, Heidelberg (2006)
19. W3C. OWL web ontology language overview, W3C Recommendation (February 10, 2004), Available at `http://www.w3.org/TR/2004/REC-owl-features-20040210/`

From DEL to EDL: Exploring the Power of Converse Events*

Guillaume Aucher and Andreas Herzig

University of Otago (NZ) - University Paul Sabatier (F)
IRIT, Université Paul Sabatier, 31062 Toulouse Cedex (F)
aucher@irit.fr, herzig@irit.fr

Abstract. Dynamic epistemic logic (DEL) as viewed by Baltag et col. and propositional dynamic logic (PDL) offer different semantics of events. On the one hand, DEL adds dynamics to epistemic logic by introducing so-called epistemic action models as syntactic objects into the language. On the other hand, PDL has instead transition relations between possible worlds. This last approach allows to easily introduce converse events. We add epistemics to this, and call the resulting logic epistemic dynamic logic (EDL). We show that DEL can be translated into EDL thanks to this use of the converse operator: this device enables us to translate the structure of the action (or event) model directly within a particular axiomatization of EDL, without having to refer to a particular epistemic action (event) model in the language (as done in DEL). It follows that EDL is more expressive and general than DEL.

1 Introduction

Aim: reason about perception of events. To account for various modes of perception of events is the aim of a family of formal systems called dynamic epistemic logics. These systems were proposed in a series of publications most prominently by Plaza, Baltag, Gerbrandy, van Benthem, van Ditmarsch, and Kooi [10,7,6,14,16,17]. Dynamic epistemic logics add dynamics to Hintikka's epistemic logic via transformations of its models.

The focus of dynamic epistemic logics is on particular events that are called updates. Updates can be seen as announcements made to the agents. The simplest case of updates are public announcements à la Plaza [10]; when the input is propositional such announcements correspond to AGM expansion operations [1]. Another example are group announcements à la Gerbrandy [6,7]. Note that DEL-updates differ from Katsuno-Mendelzon-like updates as studied in the AI literature [9].

In [2,4,3] and elsewhere, Baltag et col. proposed a dynamic epistemic logic that was very influential. We refer to it in this paper by the term DEL. It has been shown that their account subsumes all other dynamic epistemic logics,

* An extended version of this paper with proofs can be found at the following address: ftp://ftp.irit.fr/IRIT/LILAC/ecsqaruEDL.pdf

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 199–209, 2007.

justifying our acronym. The semantics of DEL is based on two kinds of models: a static model M^s (called state model by Baltag) and a (finite) dynamic model M^d (called epistemic action model by Baltag). M^s models the actual world and the agents' beliefs about it, and is nothing but a good old epistemic model à la Hintikka. M^d models the actual event taking place and the agents' beliefs about it. An agent's beliefs can be incomplete (event a occurred, but agent cannot distinguish occurrence of a from occurrence of a') and even unsound (a occurred, but agent wrongly perceived some a'). M^s and M^d are then combined by a restricted product construction which defines the situation after the actual event took place, viz. the resulting actual world, and the agents' beliefs about it.

Semantics of events: products vs. accessibility relations. Naturally, we would be interested to express in DEL that an event a occurr*ed*, viz to give semantics to the converse event a^- within the framework of DEL. It is not clear how this should work precisely. The only approach we are aware of is that of Yap [19] who fails to get a complete characterization.

On the other hand, in PDL, events are interpreted as transition relations on possible worlds, and not as restricted products of models as in DEL. Converse events a^- can then easily be interpreted by inverting the accessibility relation associated to a. The resulting logic is called the tense extension of PDL.

To this we then add an epistemic operator. We call (tensed) Epistemic Dynamic Logic EDL the combination of epistemic logic and PDL with converse.[1]

A semantics in terms of transition relations is more flexible than DEL's product semantics: we have more options concerning the interaction between events and beliefs. Our main contribution here is to account for this delicate relationship by means of constraints on the respective accessibility relations: a no-forgetting and a no-learning constraint, and a constraint of epistemic determinism.

Translating DEL into EDL. To demonstrate the power of our approach we provide a translation from DEL to EDL: we express the structure of a DEL dynamic model M^d by a nonlogical theory $\Gamma(M^d)$ of EDL, and prove that any formula φ is valid in DEL if and only if it is a logical consequence of $\Gamma(M^d)$ in EDL.

So, unlike DEL , we avoid to refer to a semantical structure (viz. the DEL dynamic model M^d) in the very definition of the language. Encoding the structure of a DEL dynamic model M^d by a nonlogical theory $\Gamma(M^d)$ of EDL is done thanks to converse events. For example $[a]B_i(\langle a^-\rangle\top \vee \langle b^-\rangle\top)$ expresses that agent i perceives the occurrence of a as that of either a or b.

Organization of the paper. This paper is organized as follows. In section 2 we introduce a language of belief, events and converse events. In section 3 we provide a semantics for that language, and define our logic EDL. In section 4 we give

[1] EDL is related to Segerberg's Doxastic Dynamic Logic DDL [12,13]. Up to now research on DDL focussed mainly on its relation with the AGM theory of belief revision, and studied particular events of the form $+\varphi$ (expansion by φ), $*\varphi$ (revision by φ), and $-\varphi$ (contraction by φ). EDL and DDL coincides in what concerns propositional announcements.

Baltag's restricted product semantics for the fragment of the language without converse, and define his logic DEL. In section 5 we associate a theory $\Gamma(M^d)$ to each dynamic model M^d, and prove that the consequences of $\Gamma(M^d)$ in EDL match the DEL-validities. This suggests that EDL is more expressive and general than DEL, and we will concentrate on that point to conclude in section 6.

2 The Languages

We suppose given sets of propositional symbols $PROP = \{p, q, \ldots\}$, agent symbols $AGT = \{i, j, \ldots\}$, and event symbols $EVT = \{a, b, \ldots\}$. All these sets may be infinite (while in DEL AGT and EVT have to be finite). From these ingredients the multi-modal language is built classically as follows:

$$\varphi := \bot | p | \neg\varphi | \varphi \wedge \varphi' | B_i\varphi | [a]\varphi | [a^-]\varphi, p \in PROP, i \in AGT, a \in EVT$$

The formula $B_i\varphi$ reads "agent i believes that φ". $[a]\varphi$ reads " φ holds after every possible occurrence of event a". $[a^-]\varphi$ reads "φ held before a". The dual modal operators $\hat{B}_i$, $\langle a\rangle$, and $\langle a^-\rangle$ are defined in the usual way: $\hat{B}_i\varphi$ abbreviates $\neg B_i\neg\varphi$; $\langle a\rangle\varphi$ abbreviates $\neg[a]\neg\varphi$; $\langle a^-\rangle\varphi$ abbreviates $\neg[a^-]\neg\varphi$.

The language $\mathcal{L}_{\mathsf{EDL}}$ of EDL is the entire language. The language $\mathcal{L}_{\mathsf{DEL}}$ of DEL is the set of those formulas of $\mathcal{L}_{\mathsf{EDL}}$ that do not contain the converse operator $[a^-]$. Finally, the epistemic language $\mathcal{L}_{EL}$ is the set of those formulas of $\mathcal{L}_{\mathsf{EDL}}$ that do not contain any dynamic operator, i.e. built from $PROP$, the Boolean operators and the B_i operators alone. For example $[a]B_i[a^-]\bot$ is an $\mathcal{L}_{\mathsf{EDL}}$-formula (that is not in $\mathcal{L}_{\mathsf{DEL}}$).

3 EDL: Epistemic Dynamic Logic with Converse

When designing models of events and beliefs the central issue is to account for the interplay of these two concepts. In our PDL-based semantics this is done by means of constraints on the respective accessibility relations. These will ensure what we call no-forgetting, no-learning and epistemic determinism.

3.1 Semantics

EDL-models are of the form

$$M = \langle W, V, \{\mathcal{A}_a\}_{a\in EVT}, \{\mathcal{B}_i\}_{i\in AGT}\rangle$$

where W is a set of possible worlds, $V : PROP \longrightarrow 2^W$ a valuation, and the $\mathcal{A}_a \subseteq W \times W$ and $\mathcal{B}_i \subseteq W \times W$ are accessibility relations on W. The relation $\mathcal{A}_a^{-1}$ is the inverse of $\mathcal{A}_a$. We sometimes view accessibility relations as mappings from worlds to sets of worlds, and write for example $\mathcal{A}_a^{-1}(w) = \{v : \langle w, v\rangle \in \mathcal{A}_a^{-1}\} = \{v : \langle v, w\rangle \in \mathcal{A}_a\}$.

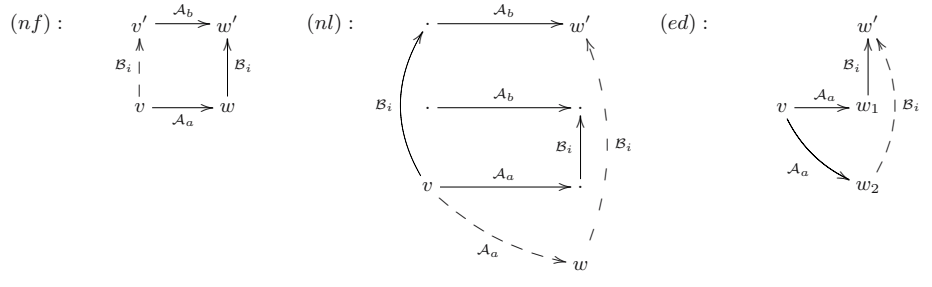
We suppose that EDL-models satisfy the following constraints of *no forgetting*, *no learning* and *epistemic determinism*:

(nf) If $v(\mathcal{A}_a \circ \mathcal{B}_i \circ \mathcal{A}_b^{-1})v'$ then $v\mathcal{B}_i v'$.
(nl) If $(\mathcal{A}_a \circ \mathcal{B}_i \circ \mathcal{A}_b^{-1})(v) \neq \emptyset$ then $(\mathcal{B}_i \circ \mathcal{A}_b)(v) \subseteq (\mathcal{A}_a \circ \mathcal{B}_i)(v)$.
(ed) If $w_1, w_2 \in \mathcal{A}_a(v)$ then $\mathcal{B}_i(w_1) = \mathcal{B}_i(w_2)$.

To understand the *no-forgetting* principle, also known as perfect recall [5], suppose that w results from the occurrence of event a in world v; if in world w, the world w' is an alternative for agent i, and w' results from event b in a world v', then v' was already possible for agent i in the world v (see figure below).

To understand the principle *no-learning*, also known as no miracle [15], assume that agent i perceive the occurrence of a as that of b_1, $b_2 \ldots$ or b_n. Then, informally, the *no-learning* principle says that *all* such alternatives resulting from occurrence of b_1, b_2,..., b_n in i's alternatives before a are indeed alternatives after a. Formally, assume that agent i perceives b as a possible alternative of a (i.e. $(\mathcal{A}_a \circ \mathcal{B}_i \circ \mathcal{A}_b^{-1})(v) \neq \emptyset$). If at v world w' was a possible outcome of event b for i, then w' is possible for i at some $w \in \mathcal{A}_a(v)$ (see figure below).

Finally, the *epistemic determinism* principle says that an agent's epistemic state after an event does not depend on the particular nondeterministic outcome. Formally, suppose we have $v\mathcal{A}_a w_1$ and $v\mathcal{A}_a w_2$. Then (ed) forces that the epistemic states at w_1 and w_2 are identical: $\mathcal{B}_i(w_1) = \mathcal{B}_i(w_2)$ (see figure below). This follows from our hypothesis that events are feedback-free (also known as uninformative events [8]): the agents cannot distinguish between their different nondeterministic outcomes. These are events of which the agents only learn their occurrence, but not their outcomes. Both public and private announcements are examples of feedback-free events. Another example is the event of tossing a coin without checking the result. An example of an event that is not feedback-free is agent i's event of testing if formula φ is true: beyond the mere occurrence of the test, i also learns about its outcome, i.e. after the test i knows whether φ is true or not. Thus the no-learning constraint is violated. Another example of a non-feedback-free event is that of tossing a coin and looking at it: here the epistemic determinism constraint is violated.



Truth of a formula φ in a world w of a model M is noted $M, w \models \varphi$ and is defined as usual:

$$M, w \models p \text{ iff } w \in V(p)$$
$$M, w \models \neg\varphi \text{ iff it is not the case that } M, w \models \varphi$$
$$M, w \models \varphi \wedge \varphi' \text{ iff } M, w \models \varphi \text{ and } M, w \models \varphi'$$

$$M, v \models B_i\varphi \text{ iff } M, v' \models \varphi \text{ for every } w' \in \mathcal{B}_i(v)$$
$$M, v \models [a]\varphi \text{ iff } M, w \models \varphi \text{ for every } w \in \mathcal{A}_a(v)$$
$$M, w \models [a^-]\varphi \text{ iff } M, v \models \varphi \text{ for every } v \in \mathcal{A}_a^{-1}(w)$$

Truth of φ in a EDL-model M is noted $M \models \varphi$ and is defined as: $M, w \models \varphi$ for every $w \in W$. Let Γ be a set of $\mathcal{L}_{\mathsf{EDL}}$-formulas. The (global) consequence relation is defined by:

$\Gamma \models_{\mathsf{EDL}} \varphi$ iff for every EDL-model M, if $M \models \psi$ for every $\psi \in \Gamma$ then $M \models \varphi$.

For example we have

$$\{[b]\varphi, \langle a\rangle B_i\langle b^-\rangle\top\} \models_{\mathsf{EDL}} [a]B_i\varphi$$

and

$$\models_{\mathsf{EDL}} (B_i[b]\varphi \wedge \langle a\rangle B_i\langle b^-\rangle\top) \rightarrow [a]B_i\varphi. \ (*)$$

Consider $\varphi = \bot$ in (*): $B_i[b]\bot$ means that perception of event b was unexpected by agent i, while $\langle a\rangle B_i\langle b^-\rangle\top$ means that i actually perceives a as b. By our no-forgetting constraint it follows that $[a]B_i\bot$ In fact, one would like to avoid agents getting inconsistent: in such situations some sort of belief revision should take place. We do not investigate this further here, and leave it to future work to augment EDL by belief revision mechanisms.

3.2 Completeness

The axiomatics of EDL is made up of the principles of multi-modal logic K for all the modal operators B_i, $[a]$ et $[a^-]$, plus the axioms (Conv_1), (Conv_2), (NF) et (NL) below:

(Conv_1) $\vdash_{\mathsf{EDL}} \varphi \rightarrow [a]\langle a^-\rangle\varphi$
(Conv_2) $\vdash_{\mathsf{EDL}} \varphi \rightarrow [a^-]\langle a\rangle\varphi$
(NF) $\vdash_{\mathsf{EDL}} B_i\varphi \rightarrow [a]B_i[b^-]\varphi$
(NL) $\vdash_{\mathsf{EDL}} \langle a\rangle\hat{B}_i\langle b^-\rangle\top \rightarrow ([a]B_i\varphi \rightarrow B_i[b]\varphi)$
(ED) $\vdash_{\mathsf{EDL}} \langle a\rangle B_i\varphi \rightarrow [a]B_i\varphi$

(Conv1) and (Conv2) are the standard conversion axioms of tense logic and converse PDL. (NF), (NL) and (ED) respectively axiomatize no forgetting, no learning and epistemic determinism.

We write $\Gamma \vdash_{\mathsf{EDL}} \varphi$ when φ is provable from the set of formulas Γ in this axiomatics.

EDL is strongly complete:

Proposition 1. *For every set of $\mathcal{L}_{EDL}$-formulas Γ and $\mathcal{L}_{EDL}$-formula φ,*

$$\Gamma \models_{EDL} \varphi \text{ if and only if } \Gamma \vdash_{EDL} \varphi.$$

Proof. The proof follows from Sahlqvist's theorem [11]: all our axioms (NF), (NL), (ED) are of the required form, and match the respective constraints (nf), (nl), (ed).

3.3 Extensions of the Basic Logic

We are going to study how our constraint (nl) evolves when we add other constraints on the epistemic accessibility relation $\mathcal{B}_i$, as it is often done in epistemic logic.

Introspective Belief. In the literature, the notion of belief is often supposed to satisfy positive and negative introspection. That is to say, the axioms of transitivity (4) and of euclidianity (5) are valid: $B_i\varphi \rightarrow B_iB_i\varphi$ (4) and $\neg B_i\varphi \rightarrow B_i\neg B_i\varphi$ (5). Semantically, in that case every $\mathcal{B}_i$ satisfies:

(45) if $w\mathcal{B}_iw'$ then $\mathcal{B}_i(w) = \mathcal{B}_i(w')$

Proposition 2. *Conditions (45), (nl), (nf) and (ed) are equivalent to conditions (45), (nl^i), (nf) and (ed) where*

(nl^i) $\mathcal{B}_i \circ \mathcal{A}_a^{-1} \circ \mathcal{B}_i \circ \mathcal{A}_a \subseteq \mathcal{B}_i$

This proposition tells us that under (45), (nl) simplifies to (nl^i). The axiom corresponding with (nl^i) is $B_i\varphi \rightarrow B_i[a^-]B_i[a]\varphi$.

Knowledge. In the literature, the notion of knowledge is often supposed to satisfy positive and negative introspection but also reflexivity (T): $B_i\varphi \rightarrow \varphi$ (T). Semantically, (T) corresponds to

(t) $w\mathcal{B}_iw$

One can then show that with this extra condition $\mathcal{B}_i$ is an equivalence relation.

Proposition 3. *Conditions (45), (t), (nl), (nf) ,(ed) are equivalent to (45), (t), (nl^t), (nf), (ed) where*

(nl^t) $\mathcal{A}_a^{-1} \circ \mathcal{B}_i \circ \mathcal{A}_a \subseteq \mathcal{B}_i$.

Proposition 3 tells us that under (45) and (t), (nl) simplifies to (nl^t). The axiom corresponding with (nl^t) is $B_i\varphi \rightarrow [a^-]B_i[a]\varphi$ and is somewhat symmetric w.r.t. axiom (NF).

4 DEL: Static Models, Dynamic Models, and Their Products

We here present the star-free version of Baltag's dynamic epistemic logic DEL [4,3].

4.1 Semantics

– ***Static models*** are just models of the form $M^s = \langle W, V, \{\overset{s}{\longrightarrow}_i\}_{i\in AGT}\rangle$, where W is an arbitrary set, $V\colon PROP \longrightarrow 2^W$ a valuation and the $\overset{s}{\longrightarrow}_i \subseteq W \times W$ are accessibility relations on W.

– ***Dynamic models*** are of the form $M^d = \langle EVT, Pre, \{\xrightarrow{d}_i\}_{i\in AGT}\rangle$, where $Pre : EVT \longrightarrow \mathcal{L}_{EL}$ is a precondition function associating epistemic formulas to events, and the $\xrightarrow{d}_i \subseteq EVT \times EVT$ are accessibility relations on EVT.

Intuitive interpretation. Informally, $Pre(a)$ is the *pre*condition that a world must fulfill so that the event a can take place in this world. For example $Pre(a) = \top$ means that action a can take place in any world. When we have $\xrightarrow{d}_i (a) = \{b\}$ then the occurrence of a is perceived by agent i as the occurrence of b; when $\xrightarrow{d}_i (a) = \{b_1, b_2\}$ then the occurrence of a is perceived by agent i indistinguishably as the occurrence of b_1 or b_2; etc.

We recall that the set EVT is the set of atomic events. In DEL it is supposed to be *finite*. Moreover, every $\xrightarrow{d}_i$ is supposed to be *serial*: for every $a \in EVT$ there is at least one $b \in EVT$ such that $a \xrightarrow{d}_i b$.

Remark 1. The basic logic DEL does not validate introspective principles. DEL can be extended as usual such that for every $i \in AGT$, the $\xrightarrow{s}_i$ and the $\xrightarrow{d}_i$ are transitive and Euclidian.

– ***Product construction.*** Given $M^s = \langle W, V, \{\xrightarrow{s}_i\}_{i\in AGT}\rangle$ and $M^d = \langle EVT, Pre, \{\xrightarrow{d}_i\}_{i\in EVT}\rangle$, their *product* $M^s \otimes M^d$ is a static model describing the situation after the event described by M^d occurred in M^s:

$$M^s \otimes M^d = \langle W', V', \{\xrightarrow{s}_i{}'\}_{i\in AGT}\rangle$$

where the new set of possible worlds is $W' = \{\langle w, a\rangle \ : \ M^s, w \models Pre(a)\}$, the new valuation is $V'(p) = \{\langle w, a\rangle \ : \ w \in V(p)\}$, and the new static accessibility relation is defined by

$$\langle w_1, a_1\rangle \xrightarrow{s}_i{}' \langle w_2, a_2\rangle \text{ iff } w_1 \xrightarrow{s}_i w_2 \text{ and } a_1 \xrightarrow{d}_i a_2.$$

– While the truth condition for the epistemic operator is just as in Hintikka's epistemic logic and in EDL, the product construction gives a semantics to the $[a]$ operator which is quite different from that of PDL and EDL:

$$M^s, w \models [a]\varphi \ \text{ iff } \ M^s, w \models Pre(a) \text{ implies } M^s \otimes M^d, \langle w, a\rangle \models \varphi$$

Finally, validity of φ in DEL (noted $\models_{\mathsf{DEL}} \varphi$) is defined as usual as truth in every world of every DEL-model. Note that validity means validity w.r.t. a fixed dynamic model M^d.

Remark 2. The truth condition for the dynamic operator highlights that DEL is a dynamic extension of epistemic logic, while EDL is an epistemic extension of PDL.

4.2 Completeness

Suppose given a dynamic model M^d. The axiomatics of DEL is made of the principles of the multi-modal logic K for the modal operators B_i and $[a]$, together with the following axioms [4,3].

(A1) $\vdash_{\mathsf{DEL}} [a]p \leftrightarrow (Pre(a) \rightarrow p)$
(A2) $\vdash_{\mathsf{DEL}} [a]\neg\varphi \leftrightarrow (Pre(a) \rightarrow \neg[a]\varphi)$
(A3) $\vdash_{\mathsf{DEL}} [a]B_i\varphi \leftrightarrow (Pre(a) \rightarrow B_i[b_1]\varphi \wedge \ldots \wedge B_i[b_n]\varphi)$
where $b_1, \ldots, b_n$ is the list of all b such that $a \xrightarrow{d}_i b$.

We note $\vdash_{\mathsf{DEL}} \varphi$ when φ is provable from these principles. Note that this axiomatization depends on a particular dynamic model M^d.

For example for every dynamic model M^d where $Pre(a) = \top$, $Pre(b) = p$, and $\xrightarrow{d}_i (a) = \{b\}$ we obtain $\vdash_{\mathsf{DEL}} [a]B_ip$. Indeed, $\vdash_{\mathsf{DEL}} [a]B_ip \leftrightarrow (Pre(a) \rightarrow B_i[b]p)$ and $\vdash_{\mathsf{DEL}} B_i[b]p$ because $\vdash_{\mathsf{DEL}} [b]p$.

5 From DEL to EDL

In this section we show that DEL can be embedded into EDL. We do that by building a particular EDL-theory that encode syntactically the structure of a given DEL dynamic model M^d.

Definition 1. *Let $M^d = \langle EVT, Pre, \{\xrightarrow{d}_i\}_{i\in AGT}\rangle$ be a dynamic model. The set of formulas $\Gamma(M^d)$ associated to M^d ('the theory of M^d') is made up of the following non-logical axioms:*

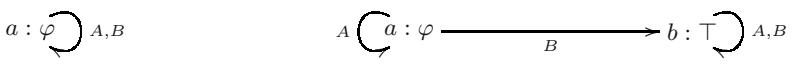
(1) $p \rightarrow [a]p$ and $\neg p \rightarrow [a]\neg p$, for every $a \in EVT$ and $p \in PROP$;
(2) $\langle a\rangle\top \leftrightarrow Pre(a)$, for every $a \in EVT$;
(3) $[a]B_i(\langle b_1^-\rangle\top \vee \ldots \vee \langle b_n^-\rangle\top)$, where $b_1, \ldots, b_n$ is the list of all b such that $a \xrightarrow{d}_i b$;
(4) $\hat{B}_i Pre(b) \rightarrow [a]\hat{B}_i\langle b^-\rangle\top$, for every $\langle a, b\rangle \in \xrightarrow{d}_i$.

Note that $\Gamma(M^d)$ is finite because in EDL both the set of events EVT and the set of agents AGT are finite.

Axiom 1 encodes the fact that events do not change propositional facts of the world where they are performed (cf the definition of $V'(p)$ in Section 4.1). Axiom 2 encodes the fact that an event a can occur in a world iff this world satisfies the precondition of event a (cf the definition of W' in Section 4.1). Axiom 3 encodes the modal structure of the dynamic model. Axiom 4 encodes the definition of $\xrightarrow{s}_i{}'$ (cf Section 4.1).

Example 1. Consider that $AGT = \{A; B\}$ and $PROP = \{p\}$. In the figure below is represented the dynamic models M_1^d and M_2^d corresponding respectively to the public announcement of φ and the private announcement of φ, where $\varphi \in \mathcal{L}_{EL}$. Here, $Pre(a) = \varphi$ in both models and $Pre(b) = \top$.

Public announcement of φ : Private announcement of φ to A:



Applying Definition 1, we get

$\Gamma(M_1^d) := \{p \rightarrow [a]p$ and $\neg p \rightarrow [a]\neg p$; $\langle a\rangle\top \leftrightarrow \varphi$; $[a]B_A(\langle a^-\rangle\top)$; $[a]B_B(\langle a^-\rangle\top)$; $\hat{B_A}\varphi \rightarrow [a]\hat{B_A}\langle a^-\rangle\top\}$; $\hat{B_B}\varphi \rightarrow [a]\hat{B_B}\langle a^-\rangle\top\}$

and

$\Gamma(M_2^d) := \{p \rightarrow [a]p$ and $\neg p \rightarrow [a]\neg p$; $p \rightarrow [b]p$ and $\neg p \rightarrow [b]\neg p$; $\langle a\rangle\top \leftrightarrow \varphi$; $\langle b\rangle\top \leftrightarrow \top$; $[a]B_A\langle a^-\rangle\top$; $[a]B_B\langle b^-\rangle\top$; $[b]B_A\langle b^-\rangle\top$; $[b]B_B\langle b^-\rangle\top$; $\hat{B_A}\varphi \rightarrow [a]\hat{B_A}\langle a^-\rangle\top$; $\hat{B_A}\top \rightarrow [b]\hat{B_A}\langle b^-\rangle\top$; $\hat{B_B}\top \rightarrow [a]\hat{B_B}\langle b^-\rangle\top$; $\hat{B_B}\top \rightarrow [b]\hat{B_B}\langle b^-\rangle\top\}$

It turns out that the axiom of determinism is a logical consequence of $\Gamma(M^d)$ in EDL . This is comforting because the axiom of determinism is indeed valid in DEL .

Lemma 1. *For every $\mathcal{L}_{DEL}$-formula φ we have $\Gamma(M^d) \models_{EDL} \langle a\rangle\varphi \rightarrow [a]\varphi$.*

Thanks to this lemma, we can now prove that for every formula φ of the language $\mathcal{L}_{\mathsf{DEL}}$, $\models_{\mathsf{DEL}} \varphi$ if and only if $\Gamma(M^d) \models_{\mathsf{EDL}} \varphi$. We first prove two lemmas.

Lemma 2. *Let M^d be a DEL dynamic model, and let ψ be a formula from $\mathcal{L}_{DEL}$. If $\not\models_{DEL} \psi$ then $\Gamma(M^d) \not\models_{EDL} \psi$.*

Lemma 3. *Let M^d be a DEL dynamic model, and let ψ be a formula from $\mathcal{L}_{DEL}$. If $\models_{DEL} \psi$ then $\Gamma(M^d) \models_{EDL} \psi$.*

Putting these two results together we obtain the following key result:

Theorem 1. *Let M^d be a DEL dynamic model. Let ψ be a formula from $\mathcal{L}_{DEL}$. Then*

$$\models_{DEL} \varphi \quad \textit{iff} \quad \Gamma(M^d) \models_{EDL} \varphi$$

It follows that

$$\vdash_{\mathsf{DEL}} \varphi \quad \text{iff} \quad \Gamma(M^d) \vdash_{\mathsf{EDL}} \varphi$$

This thus provides a new axiomatization of DEL-validities. This new axiomatization is just made of $\Gamma(M^d)$ together with the axiomatization of EDL .

6 Discussion and Conclusion

We have presented an epistemic dynamic logic EDL whose semantics differs from that of Baltag et al.'s dynamic epistemic logic DEL. We have shown that DEL can be embedded into EDL. This result allows to conclude that EDL is an

interesting alternative to Baltag et al.'s logic, that allows to talk about agents' perception of events just in the same way as DEL does. However, EDL is more expressive than DEL because it allows to talk about past events. Another of its advantages is that we can partly describe an event taking place and still draw inferences from this partial description, whereas in DEL the action (event) model has to specify everything. More generally, EDL seems more versatile than DEL to describe events. This allows to model some events that could not be modelled in DEL .

Let us demonstrate this last point by an example. Consider the situation where there are two agents i and j, and there are two possible private announcements a and b with respective preconditions p and $\neg p$. Suppose none of the agents knows anything beyond the mere fact that both a and b could have happened, i.e. $\langle a^- \rangle \top \vee \langle b^- \rangle \top$ is common knowledge. From this we should infer that the agents do not know anything about the other agent's perception (which is indeed true in reality). We can model this last fact as follows. First we recursively define the following set of formulas.

- $\Phi_i^0 = \Phi_j^0 = \{\langle a^- \rangle \top, \langle b^- \rangle \top\}$
- $\Phi_i^n = \{B_i \varphi_j : \varphi_j \in \Phi_j^{n-1}\} \cup \{\bigwedge_{\{\varphi_j : \varphi_j \in \Phi_j^{n-1}\}} \hat{B}_i \varphi_j\}$

For example we have

$$\begin{array}{l} \Phi_i^1 = \{B_i\langle a^- \rangle \top, B_i\langle b^- \rangle \top, \hat{B}_i\langle a^- \rangle \top \wedge \hat{B}_i\langle b^- \rangle \top\} \text{ and} \\ \Phi_j^2 = \{B_j B_i\langle a^- \rangle \top,\ B_j B_i\langle b^- \rangle \top,\ B_j(\hat{B}_i\langle a^- \rangle \top \wedge \hat{B}_i\langle b^- \rangle \top)\} \ \cup \\ \qquad \{\hat{B}_j B_i\langle a^- \rangle \top \wedge \hat{B}_j B_i\langle b^- \rangle \top \wedge \hat{B}_j(\hat{B}_i\langle a^- \rangle \top \wedge \hat{B}_i\langle b^- \rangle \top)\}. \end{array}$$

We naturally claim that the set of all $(\bigvee \Phi_i^n) \wedge (\bigvee \Phi_j^n)$ represents the fact that the agents do not know anything about the other agent's perception. Then we can prove by induction on n that $\{\langle a^- \rangle \top \vee \langle b^- \rangle \top\} \vdash_{\mathsf{EDL}} (\bigvee \Phi_i^n) \wedge (\bigvee \Phi_j^n)$ for every n.[2] This indicates that the agents' incomplete knowledge of what is going on is correctly represented by $\{\langle a^- \rangle \top \vee \langle b^- \rangle \top\}$. Such situations cannot be described in DEL because this would require an infinity of atomic DEL-events, and the dynamic model M^d would have to be infinite. So, in a sense, EDL seems to be more appropriate to represent situations where agents have only little cues about what is going on.

Another approach mapping DEL to automata propositional dynamic logic is [18]. He does not resort to converse events and translates dynamic models into a transformation on PDL programs. As we said in section 1, Yap introduced converse events into DEL but she failed to get a reduction axiom for the converse modal operator. Like us, she does not deal with belief revision and we leave the integration of belief revision mechanisms into EDL to further work. Another line of research is to study decidability and complexity of EDL .

[2] The key observation is that $\vdash_{\mathsf{EDL}} B_i(\langle a^- \rangle \top \vee \langle b^- \rangle \top) \rightarrow (B_i\langle a^- \rangle \top \vee B_i\langle b^- \rangle \top \vee (\hat{B}_i\langle a^- \rangle \top \wedge \hat{B}_i\langle b^- \rangle \top))$.

References

1. Alchourrón, C., GÄardenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. J. of Symbolic Logic 50, 510–530 (1985)
2. Baltag, A.: A logic of epistemic actions. Technical report, CWI (2000), http://www.cwi.nl/~abaltag/papers.html
3. Baltag, A., Moss, L.S.: Logics for epistemic programs. Synthese 139(2), 165–224 (2004)
4. Baltag, A., Moss, L.S., Solecki, S.: The logic of public announcements, common knowledge, and private suspicions. In: Proc. TARK'98, pp. 43–56. Morgan Kaufmann, San Francisco (1998)
5. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning about knowledge. MIT Press, Cambridge (1995)
6. Gerbrandy, J.: Bisimulations on Planet Kripke. PhD thesis, University of Amsterdam (1999)
7. Gerbrandy, J., Groeneveld, W.: Reasoning about information change. J. of Logic, Language and Information 6(2) (1997)
8. Herzig, A., Lang, J., Longin, D., Polacsek, T.: A logic for planning under partial observability. In: Proc. Nat (US) Conf. on Artificial Intelligence (AAAI'2000), Austin, Texas (August 2000)
9. Katsuno, H., Mendelzon, A.O.: On the di®erence between updating a knowledge base and revising it. In: GÄardenfors, P. (ed.) Belief revision, pp. 387–394. Cambridge University Press, Cambridge (1992) (preliminary version In: Allen, J.A., Fikes, R., Sandewall, E. (eds.) Principles of Knowledge Representation and Reasoning: Proc. 2nd Int. Conf., pp. 387–394. Morgan Kaufmann Publishers (1991))
10. Plaza, J.A.: Logics of public communcations. In: Emrich, M.L., Pfeifer, M.Z., Hadzikadic, M., Ras, Z.W. (eds.) Proc. 4th Int. Symposium on Methodologies for Intelligent Systems, pp. 201–216 (1989)
11. Sahlqvist, H.: Completeness and correspondence in the first and second order semantics for modal logics. In: Kanger, S. (ed.) Proc. 3rd Scandinavian Logic Symposium 1973, vol. 82, Studies in Logic, North Holland (1975)
12. Segerberg, K.: Belief revision from the point of view of doxastic logic. Bulletin of the IGPL 3, 534–553 (1995)
13. Segerberg, K.: Two traditions in the logic of belief: bringing them together. In: Ohlbach, H.J., Reyle, U. (eds.) Logic, Language and Reasoning: essays in honour of Dov Gabbay, Trends in Logic, vol. 5, pp. 135–147. Kluwer Academic Publishers, Dordrecht (1999)
14. van Benthem, J.: One is a lonely number: on the logic of communication. In: Chatzidakis, Z., Koepke, P., Pohlers, W. (eds.) Logic Colloquium'02, pp. 96–129. ASL & A.K. Peters, Wellesley MA, 2006. Tech Report PP-2002-27, ILLC Amsterdam (2002)
15. van Benthem, J., Pacuit, E.: The tree of knowledge in action: Towards a common perspective. In: Advances in Modal Logic, pp. 87–106 (2006)
16. van Ditmarsch, H.P.: Descriptions of game actions. J. of Logic, Language and Information (JoLLI) 11, 349–365 (2002)
17. van Ditmarsch, H.P., van der Hoek, W., Kooi, B.: Dynamic Epis- temic Logic. Kluwer Academic Publishers, Dordrecht (2007)
18. van Eijck, J.: Reducing dynamic epistemic logic to pdl by program transformation. Technical Report SEN-E0423, CWI (2004)
19. Yap, A.: Product update and looking backward. prepublications PP-2006-39, ILLC (2006)

Comparing Argumentation Semantics with Respect to Skepticism

Pietro Baroni and Massimiliano Giacomin

Dip. Elettronica per l'Automazione, University of Brescia,
via Branze 38, 25123 Brescia, Italy
{baroni,giacomin}@ing.unibs.it

Abstract. The issue of formalizing skepticism relations between argumentation semantics has been considered only recently in the literature. In this paper, we contribute to this kind of analysis by providing a systematic comparison of a significant set of literature semantics (namely grounded, complete, preferred, stable, semi-stable, ideal, prudent, and $CF2$ semantics) using both a weak and a strong skepticism relation.

Keywords: Argumentation semantics, Skepticism.

1 Introduction

The increasing variety of argumentation semantics proposed in the literature raises the issue of carrying out systematic principle-based comparisons between different approaches. While limitations of example-based comparisons have been pointed out earlier by several authors (see for instance [1,2]), studies on general evaluation principles for argumentation semantics are appearing in the literature only in very recent years. For instance, in [3] general *rationality postulates* for argumentation systems are introduced, showing that there are argumentation systems where they are violated. At the more abstract level of Dung's argumentation frameworks [4], in [5] several semantics evaluation criteria have been introduced and exploited for a systematic assessment of both "traditional" and more recent proposals.

In this work we consider another aspect of this kind of systematic comparison, namely the issue of (partially) ordering argumentation semantics with respect to their skepticism. After recalling the necessary background concepts in Section 2, we review in Section 3 the definitions of the *weak* and *strong* skepticism relations between semantics, first introduced in [6]. Argumentation semantics considered in this paper are quickly described in Section 4, then Section 5 shows how they are partially ordered according to the *weak* and *strong* skepticism relations. A final discussion and conclusions are provided in Section 6.

2 Basic Concepts

The present work lies in the frame of the general theory of abstract argumentation frameworks proposed by Dung [4].

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 210–221, 2007.

Definition 1. *An argumentation framework is a pair* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$*, where* $\mathcal{A}$ *is a set, and* $\rightarrow \subseteq (\mathcal{A} \times \mathcal{A})$ *is a binary relation on* $\mathcal{A}$*, called attack relation.*

In the following we will always assume that $\mathcal{A}$ is finite. Since we will frequently consider properties of sets of arguments, it is useful to extend to them the notations defined for the nodes.

Definition 2. *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$*, a node* $\alpha \in \mathcal{A}$ *and two sets* $S, P \subseteq \mathcal{A}$*, we define* $S \rightarrow \alpha \triangleq \exists \beta \in S : \beta \rightarrow \alpha$*;* $\alpha \rightarrow S \triangleq \exists \beta \in S : \alpha \rightarrow \beta$*;* $S \rightarrow P \triangleq \exists \alpha \in S, \beta \in P : \alpha \rightarrow \beta$*.*

In Dung's theory, an argumentation semantics is defined by specifying the criteria for deriving, given a generic argumentation framework, the set of all possible extensions, each one representing a set of arguments considered to be acceptable together. Accordingly, a basic requirement for any extension E is that it is *conflict-free*, namely $\nexists \alpha, \beta \in E : \alpha \rightarrow \beta$. All argumentation semantics proposed in the literature satisfy this fundamental *conflict-free property*.

Given a generic argumentation semantics $\mathcal{S}$, the set of extensions prescribed by $\mathcal{S}$ for a given argumentation framework $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ is denoted as $\mathcal{E}_{\mathcal{S}}(\mathrm{AF})$. If $\forall \mathrm{AF}\ |\mathcal{E}_{\mathcal{S}}(\mathrm{AF})| = 1$, then the semantics $\mathcal{S}$ is said to follow the *unique-status approach*, otherwise it is said to follow the *multiple-status approach*.

I-maximality is a relevant property of sets of extensions used in the following.

Definition 3. *A set of extensions* $\mathcal{E}$ *is I-maximal iff* $\forall E_1, E_2 \in \mathcal{E}$*, if* $E_1 \subseteq E_2$ *then* $E_1 = E_2$*. A semantics* $\mathcal{S}$ *satisfies the* I-maximality criterion *if and only if* $\forall \mathrm{AF}, \mathcal{E}_{\mathcal{S}}(\mathrm{AF})$ *is I-maximal.*

Note that I-maximality is a property of the set of extensions $\mathcal{E}$ *per se* and does not imply that maximality is prescribed by the semantics-specific definition of what an extension is. For instance any unique-status semantics necessarily satisfies I-maximality according to Definition 3, independently of the fact that the unique extension prescribed by the semantics is a maximal set in any sense.

It is also worth noting that it may be the case that $\mathcal{E}_{\mathcal{S}}(\mathrm{AF}) = \emptyset$, i.e. that a semantics $\mathcal{S}$ is unable to prescribe any extension for some argumentation frameworks AF. We adopt as a standpoint that such argumentation frameworks lie outside the domain of definition of $\mathcal{S}$ and therefore have not to be considered in the evaluation of its properties. Formally, for a generic semantics $\mathcal{S}$ let $\mathcal{D}_{\mathcal{S}}$ be the set of argumentation frameworks where $\mathcal{S}$ admits at least one extension, namely $\mathcal{D}_{\mathcal{S}} = \{\mathrm{AF} : \mathcal{E}_{\mathcal{S}}(\mathrm{AF}) \neq \emptyset\}$. In the following, whenever we will refer to the comparison of two semantics $\mathcal{S}_1$ and $\mathcal{S}_2$ with respect to a generic argumentation framework AF we will implicitly assume that $\mathrm{AF} \in \mathcal{D}_{\mathcal{S}_1} \cap \mathcal{D}_{\mathcal{S}_2}$. In fact, one (or even both) of the terms of comparison would be undefined otherwise.

3 Skepticism Relations

The notion of skepticism has often been used in informal ways to discuss semantics behavior, e.g. by observing that a semantics is "more skeptical" than

another one. Intuitively, a semantics is more skeptical than another if it makes less committed choices about the justification of the arguments. A comparison of skepticism between semantics can be based on a relationship $\preceq^E$ between the sets of extensions they prescribe. Given two sets of extensions $\mathcal{E}_1, \mathcal{E}_2$ of an argumentation framework AF, $\mathcal{E}_1 \preceq^E \mathcal{E}_2$ will denote that $\mathcal{E}_1$ is at least as skeptical as $\mathcal{E}_2$ in some sense. Then a relation of skepticism $\preceq^S$ between semantics, induced by $\preceq^E$, can be defined.

Definition 4. *Let $\preceq^E$ be a skepticism relation between sets of extensions. The skepticism relation between argumentation semantics $\preceq^S$ induced by $\preceq^E$ is defined as follows: for any argumentation semantics $\mathcal{S}_1$ and $\mathcal{S}_2$, $\mathcal{S}_1 \preceq^S \mathcal{S}_2 \Leftrightarrow$ for any argumentation framework* AF, $\mathcal{E}_{\mathcal{S}_1}(\mathrm{AF}) \preceq^E \mathcal{E}_{\mathcal{S}_2}(\mathrm{AF})$.

We will consider two actual skepticism relations between sets of extensions: a *weak* relation, denoted as $\preceq^E_W$, and a *strong* relation, denoted as $\preceq^E_S$. These relations have been introduced in [6] to which the reader is referred for more extensive explanations, not reported here due to space limitation. As a starting point, we recall that to compare a single extension E_1 with a set of extensions $\mathcal{E}_2$, the relation $\forall E_2 \in \mathcal{E}_2\ E_1 \subseteq E_2$ has often be used in the literature (for instance to verify that the unique extension prescribed by grounded semantics is more skeptical than the set of extensions prescribed by preferred semantics). A direct generalization to the comparison of two sets of extensions is represented by the following weak skepticism relation $\preceq^E_W$.

Definition 5. *Given two sets of extensions $\mathcal{E}_1$ and $\mathcal{E}_2$ of an argumentation framework* AF*, $\mathcal{E}_1 \preceq^E_W \mathcal{E}_2$ iff $\forall E_2 \in \mathcal{E}_2\ \exists E_1 \in \mathcal{E}_1 : E_1 \subseteq E_2$.*

Relation $\preceq^E_W$ is in a sense unidirectional, since it only constrains the extensions of $\mathcal{E}_2$, while $\mathcal{E}_1$ may contain additional extensions unrelated to those of $\mathcal{E}_2$. One may consider also a more symmetric (and stronger) relationship $\preceq^E_S$, where it is also required that any extension of $\mathcal{E}_1$ is included in an extension of $\mathcal{E}_2$.

Definition 6. *Given two sets of extensions $\mathcal{E}_1$ and $\mathcal{E}_2$ of an argumentation framework* AF*, $\mathcal{E}_1 \preceq^E_S \mathcal{E}_2$ iff $\mathcal{E}_1 \preceq^E_W \mathcal{E}_2$ and $\forall E_1 \in \mathcal{E}_1\ \exists E_2 \in \mathcal{E}_2 : E_1 \subseteq E_2$.*

By definition, given two sets of extensions $\mathcal{E}_1$ and $\mathcal{E}_2$ it holds that $\mathcal{E}_1 \preceq^E_S \mathcal{E}_2 \Rightarrow \mathcal{E}_1 \preceq^E_W \mathcal{E}_2$. Instantiating Definition 4 with $\preceq^E_W$ and $\preceq^E_S$ gives rise to two corresponding skepticism relations between semantics, denoted as $\mathcal{S}_1$ and $\mathcal{S}_2$, ordered by the same implication: $\mathcal{S}_1 \preceq^S_S \mathcal{S}_2 \Rightarrow \mathcal{S}_1 \preceq^S_W \mathcal{S}_2$.

It is worth noting that the skepticism relations introduced above are not total orders, since in general there can be two sets of extensions (and therefore two semantics) which are not comparable. We recall some properties (proved in [6]) of the skepticism relations between sets of extensions (which are of course "inherited" by the relations between semantics).

Proposition 1. *Relations $\preceq^E_W$ and $\preceq^E_S$ are preorders, i.e. they are reflexive and transitive. Relations $\preceq^E_W$ and $\preceq^E_S$ are also partial orders when the considered sets of extensions are I-maximal, namely given two I-maximal sets of extensions $\mathcal{E}_1$ and $\mathcal{E}_2$, if $\mathcal{E}_1 \preceq^E_W \mathcal{E}_2$ and $\mathcal{E}_2 \preceq^E_W \mathcal{E}_1$ then $\mathcal{E}_1 = \mathcal{E}_2$.*

4 A Review of Extension-Based Argumentation Semantics

We review the definition of several argumentation semantics which will be compared according to the skepticism relations defined in previous section.

4.1 Traditional Semantics

Stable semantics relies on the idea that an extension should be able to reject the arguments that are outside the extension itself [4].

Definition 7. *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$*, a set* $E \subseteq \mathcal{A}$ *is a* stable extension *of* AF *if and only if* E *is conflict-free* $\wedge \forall \alpha \in \mathcal{A} : \alpha \notin E,\ E \rightarrow \alpha$.

Stable semantics will be denoted as $\mathcal{ST}$, and, accordingly, the set of all the stable extensions of AF as $\mathcal{E}_{\mathcal{ST}}(\mathrm{AF})$. Stable semantics suffers by a significant limitation since there are argumentation frameworks where no extensions complying with Definition 7 exist. No other semantics considered in this paper is affected by this problem except the prudent version of stable semantics.

The requirement that an extension should attack all other external arguments can be relaxed by imposing that an extension is simply able to defend itself from external attacks. This is at the basis of the notions of *acceptable argument* and *admissible set* [4].

Definition 8. *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$*, an argument* $\alpha \in \mathcal{A}$ *is* acceptable *with respect to a set* $E \subseteq \mathcal{A}$ *if and only if* $\forall \beta \in \mathcal{A} : \beta \rightarrow \alpha,\ E \rightarrow \beta$*. Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$*, a set* $E \subseteq \mathcal{A}$ *is* admissible *if and only if* E *is conflict-free and* $\forall \beta \in \mathcal{A} : \beta \rightarrow E,\ E \rightarrow \beta$.

The set of the arguments acceptable with respect to a set E is traditionally denoted using the characteristic function $\mathrm{F}_{\mathrm{AF}}(E)$:

Definition 9. *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$*, the function* $\mathrm{F}_{\mathrm{AF}} : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ *which, given a set* $E \subseteq \mathcal{A}$*, returns the set of the acceptable arguments with respect to* E*, is called the* characteristic function *of* AF.

Building on these concepts, the notion of *complete extension* can be introduced, which plays a key role in Dung's theory, since all semantics encompassed by his framework select their extensions among the complete ones.

Definition 10. *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$*, a set* $E \subseteq \mathcal{A}$ *is a* complete extension *if and only if* E *is admissible and every argument of* $\mathcal{A}$ *which is acceptable with respect to* E *belongs to* E.

The notion of complete extension is not associated to a notion of *complete semantics* in [4], however, the term complete semantics has subsequently gained acceptance in the literature and will be used to refer to the properties of the set of complete extensions. Complete semantics will be denoted as $\mathcal{CO}$.

The well-known *grounded semantics* belongs to the unique-status approach and its unique extension, denoted as GE(AF), can be defined as the least fixed point of the characteristic function.

Definition 11. *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$*, the* grounded extension *of* AF*, denoted as* GE(AF)*, is the least fixed point (with respect to set inclusion) of* F_{AF}*.*

Preferred semantics, denoted as $\mathcal{PR}$, is obtained by simply requiring the property of maximality along with admissibility.

Definition 12. *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$*, a set* $E \subseteq \mathcal{A}$ *is a* preferred extension *of* AF *if and only if it is a maximal (with respect to set inclusion) admissible set.*

4.2 *CF*2 Semantics

$CF2$ *semantics*, first introduced in [7], is a SCC-recursive semantics [8] which features the distinctive property of treating in a "symmetric" way odd- and even-length cycles while belonging to the multiple-status approach. SCC-recursiveness is related to the graph-theoretical notion of *strongly connected components* (SCCs) of AF, namely the equivalence classes of nodes under the relation of mutual reachability, denoted as $\mathrm{SCCS}_{\mathrm{AF}}$. Due to space limitations, we can not examine in detail the definition of $CF2$ semantics: the interested reader may refer to [7] and [8].

Definition 13. *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$*, a set* $E \subseteq \mathcal{A}$ *is an extension of* $CF2$ *semantics iff*

- $E \in \mathcal{MCF}_{\mathrm{AF}}$ *if* $|\mathrm{SCCS}_{\mathrm{AF}}| = 1$
- $\forall S \in \mathrm{SCCS}_{\mathrm{AF}}\ (E \cap S) \in \mathcal{E}_{CF2}(\mathrm{AF}{\downarrow_{UP_{\mathrm{AF}}(S,E)}})$ *otherwise*

where $\mathcal{MCF}_{\mathrm{AF}}$ *denotes the set of maximal conflict-free sets of* AF*, and, for any set* $S \subseteq \mathcal{A}$*,* $\mathrm{AF}{\downarrow_S}$ *denotes the restriction of* AF *to* S*, namely* $\mathrm{AF}{\downarrow_S} = \langle S, \rightarrow \cap (S \times S) \rangle$*, and* $UP_{\mathrm{AF}}(S, E) = \{\alpha \in S \mid \nexists \beta \in E : \beta \notin S, \beta \rightarrow \alpha\}$*.*

$CF2$ semantics can be roughly regarded as selecting its extensions among the maximal conflict free sets of AF, on the basis of some topological requirements related to the decomposition of AF into strongly connected components. In particular it turns out that when AF consists of exactly one strongly connected component, the set of extensions prescribed by $CF2$ semantics exactly coincides with the set of maximal conflict free sets of AF.

4.3 Semi-stable Semantics

Semi-stable semantics [9], denoted in the following as $\mathcal{SST}$, aims at guaranteeing the existence of extensions in any case (differently from stable semantics) while coinciding with stable semantics (differently from preferred semantics) when stable extensions exist. The definition of extensions satisfying these desiderata is ingeniously simple.

Definition 14. *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ *a set* $E \subseteq \mathcal{A}$ *is a semi-stable extension if and only if* E *is a complete extension such that* $(E \cup \{\alpha \mid E \rightarrow \alpha\})$ *is maximal with respect to set inclusion.*

4.4 Ideal Semantics

Ideal semantics [10] provides an alternative unique-status approach which is less skeptical than grounded semantics, i.e. for any argumentation framework the (unique) ideal extension is a (sometimes strict) superset of the grounded extension. Also in this case the definition is quite simple.

Definition 15. *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ *a set* $E \subseteq \mathcal{A}$ *is ideal if and only if* E *is admissible and* $\forall P \in \mathcal{E}_{\mathcal{PR}}(\mathrm{AF})\ E \subseteq P$. *The ideal extension is the maximal (with respect to set inclusion) ideal set.*

We will use the symbol $\mathcal{ID}$ to refer to the ideal semantics, and denote the ideal extension of an argumentation framework AF as $ID(\mathrm{AF})$.

4.5 Prudent Semantics

Prudent semantics [11,12] emphasizes the role of indirect attacks: forbidding them leads to the definition of p(rudent)-admissible sets.

Definition 16. *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$, *an argument* α *indirectly attacks another argument* β, *denoted as* $\alpha \hookrightarrow \beta$, *if there is an odd-length path from* α *to* β *in the defeat graph corresponding to* AF. *A set* S *is without indirect conflicts, denoted as* $icf(S)$, *if and only if* $\nexists \alpha, \beta \in S : \alpha \hookrightarrow \beta$.

Definition 17. *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$, *a set of arguments* $S \subseteq \mathcal{A}$ *is p(rudent)-admissible if and only if* $\forall \alpha \in S$ α *is acceptable with respect to* S *and* $icf(S)$.

On this basis, the prudent version of several traditional notions of extensions (and then of the relevant semantics) has been defined.

Definition 18. *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$, *a set of arguments* $S \subseteq \mathcal{A}$ *is:*

- *a preferred p-extension if and only if* S *is a maximal (with respect to set inclusion) p-admissible set;*
- *a stable p-extension if and only if* $icf(S)$ *and* $\forall \alpha \in (\mathcal{A} \setminus S)\ S \rightarrow \alpha$;
- *a complete p-extension if and only if* S *is p-admissible and there is no argument* $\alpha \notin S$ *such that* α *is acceptable with respect to* S *and* $icf(S \cup \{\alpha\})$.

Definition 19. *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$, *the function* $\mathrm{F}^{p}_{\mathrm{AF}} : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ *which, given a set* $S \subseteq \mathcal{A}$, *returns the set of the acceptable arguments with respect to* S *such that* $icf(S \cup \{\alpha\})$ *is called the* p-characteristic function *of* AF. *Let* j *be the lowest integer such that the sequence* $\mathrm{F}^{p,i}_{\mathrm{AF}}(\emptyset)$ *is stationary for* $i \geq j$: $\mathrm{F}^{p,j}_{\mathrm{AF}}(\emptyset)$ *is the grounded p-extension of* AF, *denoted as* $\mathrm{GPE}(\mathrm{AF})$.

The prudent versions of grounded, complete, preferred and stable semantics will be denoted as $\mathcal{GRP}$, $\mathcal{COP}$, $\mathcal{PRP}$ and $\mathcal{STP}$, respectively.

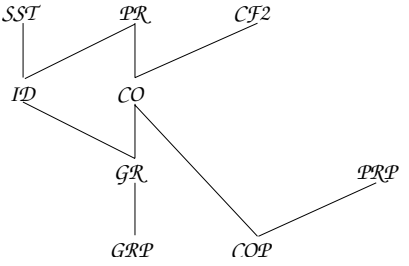


Fig. 1. $\preceq_S^S$ relation for any argumentation framework

5 Skepticism Comparison of Argumentation Semantics

In this section we examine comparability between semantics according to the relations $\preceq_W^S$ and $\preceq_S^S$. Before entering the matter, a result which will be used in the following and whose proof is immediate needs to be stated.

Lemma 1. *Given two argumentation semantics* $\mathcal{S}_1$ *and* $\mathcal{S}_2$*, if for any argumentation framework* AF $\mathcal{E}_{\mathcal{S}2}(\mathrm{AF}) \subseteq \mathcal{E}_{\mathcal{S}1}(\mathrm{AF})$*, then* $\mathcal{S}1 \preceq_W^S \mathcal{S}2$*.*

Note also from Proposition 1 that, given two distinct semantics $\mathcal{S}_1$, $\mathcal{S}_2$ (i.e. such that $\exists \mathrm{AF} : \mathcal{E}_{\mathcal{S}_1}(\mathrm{AF}) \neq \mathcal{E}_{\mathcal{S}_2}(\mathrm{AF})$) satisfying the I-maximality criterion, if $\mathcal{S}1 \preceq_W^S \mathcal{S}2$ then $\mathcal{S}2 \not\preceq_W^S \mathcal{S}1$ (and also $\mathcal{S}2 \not\preceq_S^S \mathcal{S}1$). Since all semantics reviewed in Section 4 are distinct and, with the exception of $\mathcal{CO}$ and $\mathcal{COP}$, also I-maximal, this fact will be implicitly exploited for relations not involving $\mathcal{CO}$ and $\mathcal{COP}$.

To begin our comparison, we examine $\preceq_S^S$ whose Hasse diagram is shown in Figure 1. Starting from the bottom, let us show that $\mathcal{GRP} \preceq_S^S \mathcal{GR}$, which, both belonging to the unique-status approach, is equivalent to the inclusion relation proved in Proposition 2.

Proposition 2. *For any argumentation framework* AF*,* $\mathrm{GPE}(\mathrm{AF}) \subseteq \mathrm{GE}(\mathrm{AF})$*.*

Proof. Recall from [4] that $\mathrm{GE}(\mathrm{AF}) = \bigcup_{i \geq 1} \mathrm{F}^i_{\mathrm{AF}}(\emptyset)$ and $\mathrm{GPE}(\mathrm{AF}) = \mathrm{F}^{p,j}_{\mathrm{AF}}(\emptyset)$, where j is the lowest integer such that the sequence $\mathrm{F}^{p,i}_{\mathrm{AF}}(\emptyset)$ is stationary for $i \geq j$. Now, obviously $\mathrm{F}^1_{\mathrm{AF}}(\emptyset) = \mathrm{F}^{p,1}_{\mathrm{AF}}(\emptyset)$ (since both coincide with the set of unattacked arguments in AF). Assume inductively that $\mathrm{F}^{p,i}_{\mathrm{AF}}(\emptyset) \subseteq \mathrm{F}^i_{\mathrm{AF}}(\emptyset)$, then it holds that $\mathrm{F}^{p,i+1}_{\mathrm{AF}}(\emptyset) \subseteq \mathrm{F}^{i+1}_{\mathrm{AF}}(\emptyset)$. In fact, any $\alpha \in \mathrm{F}^{p,i+1}_{\mathrm{AF}}(\emptyset)$ is defended by $\mathrm{F}^{p,i}_{\mathrm{AF}}(\emptyset)$, but then it is also defended by $\mathrm{F}^i_{\mathrm{AF}}(\emptyset) \supseteq \mathrm{F}^{p,i}_{\mathrm{AF}}(\emptyset)$, and therefore $\alpha \in \mathrm{F}^{i+1}_{\mathrm{AF}}(\emptyset)$.

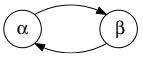


Fig. 2. A mutual attack

Going up in the diagram, since, as well-known, the grounded extension is included in any complete extension, $\mathcal{GR} \preceq_S^S \mathcal{CO}$. On the other hand, $\mathcal{CO} \not\preceq_S^S \mathcal{GR}$, as shown, for instance, by the example of Figure 2 where $\mathcal{E}_{\mathcal{CO}}(\text{AF}) = \{\emptyset, \{\alpha\}, \{\beta\}\}$, while $\mathcal{E}_{\mathcal{GR}}(\text{AF}) = \{\emptyset\}$. Furthermore, it is easy to see that $\mathcal{CO} \preceq_S^S \mathcal{PR}$ since any preferred extension is also a complete extension and any complete extension, being an admissible set, is included in a maximal admissible set, i.e. in a preferred extension. On the other hand, $\mathcal{PR} \not\preceq_S^S \mathcal{CO}$ (indeed $\mathcal{PR} \not\preceq_W^S \mathcal{CO}$): considering again Figure 2 we have $\mathcal{E}_{\mathcal{PR}}(\text{AF}) = \{\{\alpha\}, \{\beta\}\}$, $\mathcal{E}_{\mathcal{CO}}(\text{AF}) = \{\emptyset, \{\alpha\}, \{\beta\}\}$ and there is no preferred extension E_1 such that $E_1 \subseteq \emptyset \in \mathcal{E}_{\mathcal{CO}}(\text{AF})$.

As to the upper-left part of the diagram, it is shown in [10] that $\text{GE}(\text{AF}) \subseteq ID(\text{AF})$, entailing that $\mathcal{GR} \preceq_S^S \mathcal{ID}$. Moreover, by definition, the ideal extension is included in any preferred extension. It follows that $\mathcal{ID} \preceq_S^S \mathcal{PR}$ and, since any semi-stable extension is also a preferred extension [9], $\mathcal{ID} \preceq_S^S \mathcal{SST}$.

As to $CF2$ semantics, it is known [8] that $\forall \text{AF}\ \forall E \in \mathcal{E}_{CF2}(\text{AF})\ \text{GE}(\text{AF}) \subseteq E$, which, since $\text{GE}(\text{AF}) \in \mathcal{E}_{\mathcal{CO}}(\text{AF})$ entails $\mathcal{CO} \preceq_W^S CF2$. Moreover, it is known [7] that $\forall E_1' \in \mathcal{E}_{\mathcal{PR}}(\text{AF})\ \exists E_2 \in \mathcal{E}_{CF2}(\text{AF}) : E_1' \subseteq E_2$. Since in turn any complete extension is included in a preferred extension, summing up we have that $\mathcal{CO} \preceq_S^S CF2$. On the other hand, since in the example of Figure 2 $CF2$ semantics behaves as preferred semantics, it follows that $CF2 \not\preceq_S^S \mathcal{CO}$ (indeed $CF2 \not\preceq_W^S \mathcal{CO}$).

Turning to the right bottom part, it is easy to see that $\mathcal{COP} \preceq_S^S \mathcal{CO}$. $\mathcal{COP} \preceq_W^S \mathcal{CO}$ follows from the fact that $\text{GPE}(\text{AF}) \in \mathcal{E}_{\mathcal{COP}}(\text{AF})$ and $\forall E_2 \in \mathcal{E}_{\mathcal{CO}}(\text{AF})$ $\text{GPE}(\text{AF}) \subseteq \text{GE}(\text{AF}) \subseteq E_2$ (using Proposition 2 for the first inclusion). Moreover, any complete prudent extension is an admissible set and is therefore included in a preferred extension, which is also a complete extension of AF. On the other hand, $\mathcal{CO} \not\preceq_S^S \mathcal{COP}$ (indeed $\mathcal{CO} \not\preceq_W^S \mathcal{COP}$): considering Figure 3, we have $\mathcal{E}_{\mathcal{COP}}(\text{AF}) = \{\{\delta, \epsilon\}, \{\alpha, \epsilon\}\}$ (note in particular that α indirectly conflicts with δ), while $\mathcal{E}_{\mathcal{CO}}(\text{AF}) = \{\{\delta, \epsilon, \alpha\}\}$.

Turning finally to preferred prudent semantics, since any preferred prudent extension is also a complete prudent extension and any complete prudent extension is included in a preferred prudent extension, it holds that $\mathcal{COP} \preceq_S^S \mathcal{PRP}$. Since in the example of Figure 2 complete prudent and preferred prudent behave as their traditional counterparts, it follows that $\mathcal{PRP} \not\preceq_S^S \mathcal{COP}$ (indeed $\mathcal{PRP} \not\preceq_W^S \mathcal{COP}$).

Let us consider now the weak skepticism relation $\preceq_W^S$, whose Hasse diagram is shown in Figure 4: we will comment only on edges not implied by the relations $\preceq_S^S$ already examined. Starting from the bottom, since $\text{GPE}(\text{AF}) \in \mathcal{E}_{\mathcal{COP}}(\text{AF})$ Lemma 1 directly entails that $\mathcal{COP} \preceq_W^S \mathcal{GRP}$. On the other hand, considering Figure 3 and recalling that $\mathcal{E}_{\mathcal{COP}}(\text{AF}) = \{\{\delta, \epsilon\}, \{\alpha, \epsilon\}\}$, we have that

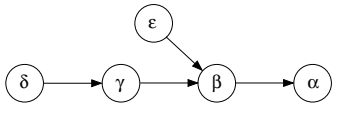


Fig. 3. Direct and indirect attacks

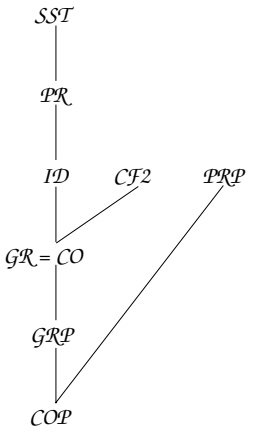


Fig. 4. $\preceq_W^S$ relation for any argumentation framework

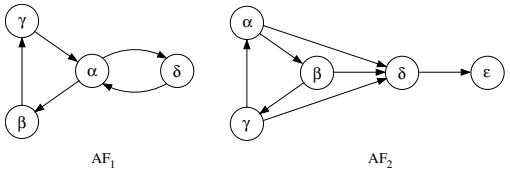


Fig. 5. $CF2$ semantics is not comparable with some others

$\mathrm{GPE}(\mathrm{AF}) = \{\{\delta, \epsilon\}\}$, which entails $\mathcal{GRP} \not\preceq_W^S \mathcal{COP}$. The next difference with respect to $\preceq_S^S$ concerns grounded and complete semantics. We already know that $\mathcal{GR} \preceq_W^S \mathcal{CO}$, we now note, by Lemma 1, that $\mathcal{CO} \preceq_W^S \mathcal{GR}$, since $\mathrm{GE}(\mathrm{AF}) \in \mathcal{E}_{\mathcal{CO}}(\mathrm{AF})$. Finally, again by Lemma 1, we have $\mathcal{PR} \preceq_W^S \mathcal{SST}$ since it is shown in [9] that any semi-stable extension is also a preferred extension.

While we have now proved the existence of all the edges shown in Figures 1 and 4, one might wonder whether additional relations hold. We prove that this is not the case, starting from $\preceq_W^S$ relation. Consider first $CF2$ semantics: it is not comparable with ideal, preferred and semi-stable semantics. In fact, referring to Figure 5, it holds that $\mathcal{E}_{\mathcal{PR}}(\mathrm{AF}_1) = \mathcal{E}_{\mathcal{SST}}(\mathrm{AF}_1) = \mathcal{E}_{\mathcal{ID}}(\mathrm{AF}_1) = \{\{\beta, \delta\}\}$ while $\mathcal{E}_{CF2}(\mathrm{AF}_1) = \{\{\gamma, \delta\}, \{\beta, \delta\}, \{\alpha\}\}$, therefore letting $\mathcal{S}$ be any of the three considered semantics $\mathcal{S} \not\preceq_W^S CF2$. On the other hand, it holds that $\mathcal{E}_{\mathcal{PR}}(\mathrm{AF}_2) = \mathcal{E}_{\mathcal{SST}}(\mathrm{AF}_2) = \mathcal{E}_{\mathcal{ID}}(\mathrm{AF}_2) = \{\emptyset\}$, while $\mathcal{E}_{CF2}(\mathrm{AF}_2) = \{\{\alpha, \epsilon\}, \{\beta, \epsilon\}, \{\gamma, \epsilon\}\}$, from which it follows that $CF2 \not\preceq_W^S \mathcal{S}$.

Turning to preferred prudent semantics, by transitivity of $\preceq_W^S$ it is sufficient to show $\mathcal{GRP} \not\preceq_W^S \mathcal{PRP}$, $\mathcal{PRP} \not\preceq_W^S \mathcal{SST}$ and $\mathcal{PRP} \not\preceq_W^S CF2$. As to the first condition, in the example of Figure 3 it holds that $\mathcal{E}_{\mathcal{PRP}}(\mathrm{AF}) = \{\{\delta, \epsilon\}, \{\alpha, \epsilon\}\}$ while $\mathrm{GPE}(\mathrm{AF}) = \{\delta, \epsilon\}$. Then, letting $\mathcal{S}$ be any of the semantics shown in Figure 4 (except the complete prudent semantics), it holds that $\mathcal{S} \not\preceq_W^S \mathcal{PRP}$. As to the second and third condition, in the example of Figure 6 taken from [12] it holds that $\mathcal{E}_{\mathcal{PRP}}(\mathrm{AF}) = \{\{\eta\}\}$, while $\mathcal{E}_{\mathcal{SST}}(\mathrm{AF}) = \{\{\alpha, \gamma, \eta\}, \{\beta, \delta, \zeta\}, \{\beta, \delta, \eta\}\}$

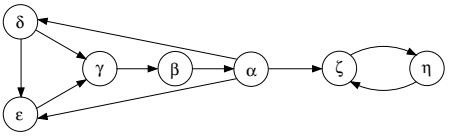


Fig. 6. Preferred prudent semantics is not comparable with many others

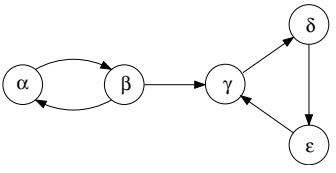


Fig. 7. A case showing that $\mathcal{COP} \not\preceq_S^S \mathcal{SST}$

and $\mathcal{E}_{CF2}(\text{AF}) = \{\{\alpha, \gamma, \eta\}, \{\beta, \delta, \zeta\}, \{\beta, \delta, \eta\}, \{\epsilon, \beta, \eta\}, \{\epsilon, \beta, \zeta\}\}$. By transitivity, second condition implies $\mathcal{PRP} \not\preceq_W^S \mathcal{S}$ for $\mathcal{S} \in \{\mathcal{GRP}, \mathcal{GR}, \mathcal{CO}, \mathcal{ID}, \mathcal{PR}\}$.

Turning to non-existence of edges in Figure 1, first note that $\mathcal{S}_1 \not\preceq_W^S \mathcal{S}_2 \Rightarrow \mathcal{S}_1 \not\preceq_S^S \mathcal{S}_2$. In particular, nothing remains to be said about $CF2$ and $\mathcal{PRP}$ semantics. As to $\mathcal{COP}$ semantics, the example of Figure 7 shows that $\mathcal{COP} \not\preceq_S^S \mathcal{SST}$ since $\mathcal{E}_{\mathcal{SST}}(\text{AF}) = \{\{\beta, \delta\}\}$, while there is a complete prudent extension, namely $\{\alpha\}$, which is not included in any semi-stable extension. By transitivity of $\preceq_S^S$, this also entails $\mathcal{COP} \not\preceq_S^S \mathcal{S}$ for any $\mathcal{S} \in \{\mathcal{GRP}, \mathcal{GR}, \mathcal{ID}\}$. On the other hand, for any semantics $\mathcal{S} \in \{\mathcal{GRP}, \mathcal{GR}, \mathcal{ID}, \mathcal{SST}\}$ we already know from Figure 4 that $\mathcal{S}$ is not comparable with $\mathcal{PRP}$, which entails $\mathcal{S} \not\preceq_S^S \mathcal{COP}$. As to preferred semantics, we already know that $\mathcal{SST} \not\preceq_S^S \mathcal{PR}$, while $\mathcal{PR} \not\preceq_S^S \mathcal{SST}$ holds since otherwise, by transitivity, it would be the case that $\mathcal{COP} \preceq_S^S \mathcal{SST}$. Similarly,

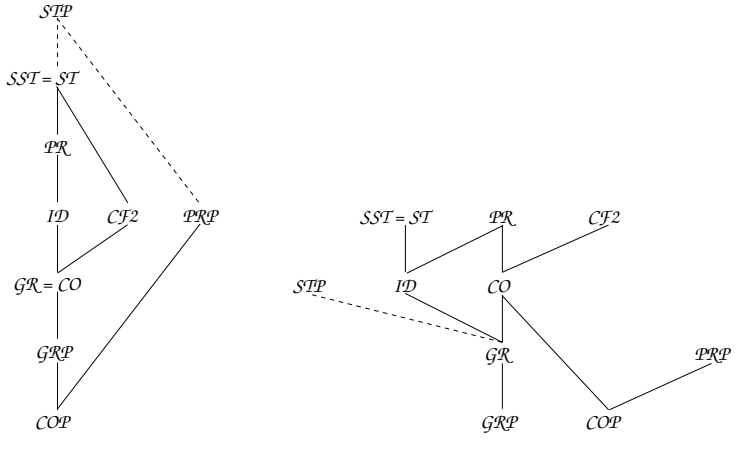


Fig. 8. $\preceq_W^S$ and $\preceq_S^S$ for argumentation frameworks where stable extensions exists

from Figure 4 we have $\mathcal{ID} \not\preceq_S^S \mathcal{CO}$ and $\mathcal{SST} \not\preceq_S^S \mathcal{CO}$. Furthermore, $\mathcal{COP} \not\preceq_S^S \mathcal{SST}$ implies $\mathcal{CO} \not\preceq_S^S \mathcal{ID}$ and $\mathcal{CO} \not\preceq_S^S \mathcal{SST}$.

Having completed the analysis concerning argumentation semantics able to prescribe extensions in any case, in Figure 8 we provide (without comments and proofs due to space limitation) the Hasse diagrams restricted to the case of argumentation frameworks where stable extensions exist (relations concerning stable prudent semantics, when its extensions exist, are shown dashed).

6 Conclusions

We have provided a systematic skepticism comparison concerning a significant range of both "traditional" and more recent argumentation semantics, using both a weak and a strong comparison criterion. The weak criterion gives rise to an almost linear ordering not including just $CF2$ and preferred prudent semantics. Semantics related to the notion of stable extension, namely semi-stable, stable and stable prudent turn out to be the least skeptical, while the notions of grounded and complete extension (more so their prudent counterpart) provide a bottom reference for skepticism. Ideal and preferred semantics lie orderly between grounded and stable-related semantics and can be regarded as "intermediate". So does $CF2$ semantics, while being not comparable with ideal and preferred. It is also interesting to note that prudent versions of stable, grounded and complete semantics tend to make "more extreme" the skepticism properties of their traditional counterparts, while preferred prudent semantics shows a sort of singularity being comparable only with complete prudent and stable prudent semantics. The strong relation gives rise to a more complicated situation, where grounded and complete semantics are not equivalent any more while grounded prudent and complete prudent semantics are still the most skeptical in some sense (but are incomparable each other). At the "top level", stable-related, preferred, and $CF2$ semantics are not comparable and turn out to be less skeptical than any other semantics they are comparable with. Ideal and complete semantics play a sort of intermediate role between grounded and other less skeptical semantics, while preferred prudent semantics is still somehow isolated.

While all the above remarks are interesting in some respect, one may be led to conclude that it is probably the case that the strong relation is actually too demanding (as also observed when applying these criteria to a different kind of analysis in [13]), while the weak relation is more reasonable and gives rise, as a consequence, to a more useful picture. Different pictures would be obtained considering alternative notions of skepticism, which is a topic for future work. It has however been proved in [14] that comparing the intersection of all extensions gives rise to the same partial order as $\preceq_W^S$. Finally, it has to be remarked that skepticism can be regarded as an attitude rather than an evaluation criterion for semantics: a more (or less) skeptical semantics is not preferable *per se*. In fact, characterizing the appropriate level of skepticism with respect to the requirements of a specific reasoning context is an interesting open problem: an example of this kind of investigation, concerning epistemic vs. practical reasoning, is given in [15].

Acknowledgments. We thank the referees for their helpful comments.

References

1. Vreeswijk, G.A.W.: Interpolation of benchmark problems in defeasible reasoning. In: Proc. of the 2nd World Conf. on the Fundamentals of Artificial Intelligence (WOCFAI 95), Paris, France, pp. 453–468 (1995)
2. Prakken, H.: Intuitions and the modelling of defeasible reasoning: some case studies. In: Proc. of the 9th Int. Workshop on Non-Monotonic Reasoning (NMR 2002), Toulouse, France, pp. 91–102 (2002)
3. Caminada, M., Amgoud, L.: An axiomatic account of formal argumentation. In: Proc. of the Twentieth National Conf. on Artificial Intelligence (AAAI-05), Menlo Park, pp. 608–613. AAAI Press, Stanford (2005)
4. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games. Artificial Intelligence 77(2), 321–357 (1995)
5. Baroni, P., Giacomin, M.: Evaluation and comparison criteria for extension-based argumentation semantics. In: Proc. of the 1st Int. Conf. on Computational Models of Arguments (COMMA 2006), Liverpool, pp. 157–168. IOS Press, Amsterdam (2006)
6. Baroni, P., Giacomin, M., Guida, G.: Towards a formalization of skepticism in extension-based argumentation semantics. In: Proc. of the 4th Workshop on Computational Models of Natural Argument (CMNA 2004), Valencia, E, pp. 47–52 (2004)
7. Baroni, P., Giacomin, M.: Solving semantic problems with odd-length cycles in argumentation. In: Nielsen, T.D., Zhang, N.L. (eds.) ECSQARU 2003. LNCS (LNAI), vol. 2711, pp. 440–451. Springer, Heidelberg (2003)
8. Baroni, P., Giacomin, M., Guida, G.: SCC-recursiveness: a general schema for argumentation semantics. Artificial Intelligence 168(1-2), 165–210 (2005)
9. Caminada, M.: Semi-stable semantics. In: Proc. of the 1st Int. Conf. on Computational Models of Arguments (COMMA 2006), Liverpool, pp. 121–130. IOS Press, Amsterdam (2006)
10. Dung, P.M., Mancarella, P., Toni, F.: A dialectic procedure for sceptical, assumption-based argumentation. In: Proc. of the 1st Int. Conf. on Computational Models of Arguments (COMMA 2006), Liverpool, pp. 145–156. IOS Press, Amsterdam (2006)
11. Coste-Marquis, S., Devred, C., Marquis, P.: Prudent semantics for argumentation frameworks. In: Proc. of the 17th IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI 2005), Hong Kong, pp. 568–572. IEEE Computer Society Press, Los Alamitos (2005)
12. Coste-Marquis, S., Devred, C., Marquis, P.: Sémantiques prudentes pour les systèmes d'argumentation. In: Proc. of the 15th Congres AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA 2006), Tours, F (2006)
13. Baroni, P., Giacomin, M.: Evaluating argumentation semantics with respect to skepticism adequacy. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 329–340. Springer, Heidelberg (2005)
14. Baroni, P., Giacomin, M.: On principle-based evaluation, comparison, and design of extension-based argumentation semantics. Tech. Rep., Univ. of Brescia, I (2007)
15. Prakken, H.: Combining sceptical epistemic reasoning with credulous practical reasoning. In: Proc. of the 1st Int. Conf. on Computational Models of Arguments (COMMA 2006), Liverpool, pp. 311–322. IOS Press, Amsterdam (2006)

An Algorithm for Computing Semi-stable Semantics*

Martin Caminada

Utrecht University / University of Luxembourg

Abstract. The semi-stable semantics for formal argumentation has been introduced as a way of approximating stable semantics in situations where no stable extensions exist. Semi-stable semantics can be located between stable semantics and preferred semantics in the sense that every stable extension is a semi-stable extension and every semi-stable extension is a preferred extension. Moreover, in situations where at least one stable extension exists, the semi-stable extensions are equal to the stable extensions. In this paper we provide an outline of an algorithm for computing the semi-stable extensions, given an argumentation framework. We show that with a few modifications, the algorithm can also be used for computing stable and preferred semantics.

1 Introduction

Formal argumentation, as a technique for defeasible entailment, has gained popularity since it combines a relatively easy to understand and human-style approach to reasoning with the mathematical rigidness that is required for software implementation [1]. It is also an interesting observation that many formalisms for nonmonotonic reasoning can be expressed as instances of formal argumentation [2].

Formal argumentation, in its most abstract form, is done using a set of abstract arguments and a defeat relation between these arguments. Since an argument A may be defeated by another argument B which may in its turn be defeated by a third argument C, the status of A (whether it can be accepted or not) partly depends on the status of C. Thus, what is needed is an overall criterion for determining which of the arguments can be considered to be ultimately justified. Several of such criteria have been proposed, the most well-known of these are grounded, preferred and stable semantics [2]. A relatively new proposal is semi-stable semantics [3]. Semi-stable semantics can be placed between stable semantics and preferred semantics, as every stable extension is also a semi-stable extension and every semi-stable extension is also a preferred extension. Moreover, semi-stable semantics can be seen as a way of approximating stable semantics in situations where no stable extensions exist.

* This work was sponsored by the Netherlands Organization for Scientific Research (NWO). We also thank Newres al Haider for helping to develop the initial idea.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 222–234, 2007.

In this paper we present an algorithm for computing all semi-stable extensions of an argumentation framework. In order to keep the discussion brief, full formal proofs are provided in a seperate technical report [4].

2 Argument-Based Semantics

In this section, we provide a brief introduction on argument based semantics and the position of semi-stable semantics.

Definition 1. *An* argumentation framework *is a pair* (Ar, def) *where* Ar *is a finite set of arguments and* $def \subseteq Ar \times Ar$.

We say that an argument A *defeats* an argument B iff $(A, B) \in def$.

An argumentation framework can be represented as a directed graph in which the arguments are represented as nodes and the defeat relation is represented as arrows. In several examples throughout this paper, we will use this graph representation.

The shorthand notation A^+ and A^- stands for, respectively, the set of arguments defeated by A and the set of arguments that defeat A. Likewise, if $\mathcal{A}rgs$ is a set of arguments, then we write $\mathcal{A}rgs^+$ for the set of arguments that is defeated by at least one argument in $\mathcal{A}rgs$, and $\mathcal{A}rgs^-$ for the set of arguments that defeat at least one argument in $\mathcal{A}rgs$. In the definition below, $F(\mathcal{A}rgs)$ stands for the set of arguments that are acceptable in the sense of [2].

Definition 2 (defense / conflict-free). *Let* $A \in Ar$ *and* $\mathcal{A}rgs \subseteq Ar$.
We define A^+ *as* $\{B \mid A \text{ def } B\}$ *and* $\mathcal{A}rgs^+$ *as* $\{B \mid A \text{ def } B \text{ with } A \in \mathcal{A}rgs\}$.
We define A^- *as* $\{B \mid B \text{ def } A\}$ *and* $\mathcal{A}rgs^-$ *as* $\{B \mid B \text{ def } A \text{ with } A \in \mathcal{A}rgs\}$.
$\mathcal{A}rgs$ *is* conflict-free *iff* $\mathcal{A}rgs \cap \mathcal{A}rgs^+ = \emptyset$. $\mathcal{A}rgs$ defends *an argument* A *iff* $A^- \subseteq \mathcal{A}rgs^+$. *We define the function* $F : 2^{Ar} \rightarrow 2^{Ar}$ *as* $F(\mathcal{A}rgs) = \{A \mid A \text{ is defended by } \mathcal{A}rgs\}$.

In the definition below, definitions of grounded, preferred and stable semantics are described in terms of complete semantics, which has the advantage of making the proofs in the remainder of this paper more straightforward. These descriptions are not literally the same as the ones provided by Dung [2], but as was first stated in [5], these are in fact equivalent to Dung's original versions of grounded, preferred and stable semantics.

Definition 3 (acceptability semantics). *A conflict-free set* $\mathcal{A}rgs$ *of arguments is called*

- *an* admissible *set iff* $\mathcal{A}rgs \subseteq F(\mathcal{A}rgs)$.
- *a* complete *extension iff* $\mathcal{A}rgs = F(\mathcal{A}rgs)$.
- *a* grounded *extension iff* $\mathcal{A}rgs$ *is the minimal complete extension.*
- *a* preferred *extension iff* $\mathcal{A}rgs$ *is a maximal complete extension.*
- *a* stable *extension iff* $\mathcal{A}rgs$ *is a complete extension that defeats every argument in* $Ar \backslash \mathcal{A}rgs$.

- *a* semi-stable *extension iff* $\mathcal{A}rgs$ *is a complete extension where* $\mathcal{A}rgs \cup \mathcal{A}rgs^+$ *is maximal (w.r.t. set-inclusion)*

In [3] it is proved that every stable extension is also a semi-stable extension, and the every semi-stable extension is also a preferred extension. Moreover, it is observed that if the argumentation framework has at least one stable extension, then the set of semi-stable extensions is equal to the set of stable extensions. That is, when at least one stable extension exists, then stable semantics and semi-stable semantics coincide.

3 A Brief Introduction to Argument Labellings

The concepts of admissibility, as well as those of complete, grounded, preferred, stable or semi-stable semantics were originally stated in terms of sets of arguments. It is equally well possible, however, to express these concepts using *argument labellings*. This approach was originally proposed by Pollock [6] and has recently been extended by Caminada [5]. The idea of a labelling is to associate with each argument exactly one label, which can either be `in`, `out` or `undec`. The label `in` indicates that the argument is explicitly accepted, the label `out` indicates that the argument is explicitly rejected, and the label `undec` indicates that the status of the argument is undecided, meaning that one abstains from an explicit judgement whether the argument is `in` or `out`.

Definition 4. *A* labelling *is a function* $\mathcal{L} : Ar \longrightarrow \{\texttt{in}, \texttt{out}, \texttt{undec}\}$.

We write $\texttt{in}(\mathcal{L})$ for $\{A \mid \mathcal{L}(A) = \texttt{in}\}$, $\texttt{out}(\mathcal{L})$ for $\{A \mid \mathcal{L}(A) = \texttt{out}\}$ and $\texttt{undec}(\mathcal{L})$ for $\{A \mid \mathcal{L}(A) = \texttt{undec}\}$. Sometimes, we write a labelling $\mathcal{L}$ as a triple $(\mathcal{A}rgs_1, \mathcal{A}rgs_2, \mathcal{A}rgs_3)$ where $\mathcal{A}rgs_1 = \texttt{in}(\mathcal{L})$, $\mathcal{A}rgs_2 = \texttt{out}(\mathcal{L})$ and $\mathcal{A}rgs_3 = \texttt{undec}(\mathcal{L})$. We distinguish three special kinds of labellings. The *all-in labelling* is a labelling that labels every argument `in`. The *all-out labelling* is a labelling that labels every argument `out`. The *all-undec labelling* is a labelling that labels every argument `undec`.

Definition 5. *Let* $\mathcal{L}$ *be a labelling and A be an argument. We say that:*

1. *A is* illegally `in` *iff A is labelled* `in` *but not all its defeaters are labelled* `out`
2. *A is* illegally `out` *iff A is labelled* `out` *but does not have a defeater labelled* `in`
3. *A is* illegally `undec` *iff A is labelled* `undec` *but either all its defeaters are labelled* `out` *or it has a defeater that is labelled* `in`.

We say that a labelling has no illegal arguments *iff there is no argument that is illegally* `in`*, illegally* `out` *or illegally* `undec`*. We say that an argument is* legally `in` *iff it is labelled* `in` *and is not illegally* `in`*. We say that an argument is* legally `out` *iff it is labelled* `out` *and is not illegally* `out`*. We say that an argument is* legally `undec` *iff it is labelled undec and is not illegally undec.*

Definition 6. *An* admissible labelling *is a labelling without arguments that are illegally* `in` *and without arguments that are illegally* `out`*.*

Definition 7. *A* complete labelling *is a labelling without arguments that are illegally* `in`*, without arguments that are illegally* `out` *and without arguments that are illegally* `undec`*.*

Definition 8. *Let $\mathcal{L}$ be a complete labelling. We say that $\mathcal{L}$ is a*

- grounded labelling *iff* $\texttt{in}(\mathcal{L})$ *is minimal (w.r.t. set inclusion).*
- preferred labelling *iff* $\texttt{in}(\mathcal{L})$ *is maximal (w.r.t. set inclusion).*
- stable labelling *iff* $\texttt{undec}(\mathcal{L}) = \emptyset$*.*
- semi-stable labelling *iff* $\texttt{undec}(\mathcal{L})$ *is minimal (w.r.t. set inclusion).*

As an illustration of how the various types of labellings can be applied, consider the two examples in Figure 1. For the example at the left hand side of Figure 1, there exists just one complete labelling: $(\{B, D\}, \{C\}, \{A\})$, which is then automatically also grounded, preferred and semi-stable. The example at the left hand side does not have any stable labellings. For the example at the right hand side of Figure 1, there exist three complete labellings: $(\emptyset, \emptyset, \{A, B, C, D, E\})$, $(\{A\}, \{B\}, \{C, D, E\})$ and $(\{B, D\}, \{A, C, E\}, \emptyset)$. The first labelling is the grounded labelling. The second and third labellings are both preferred labellings. The third labelling is also a stable and semi-stable labelling.

As for the admissible labellings, it should be mentioned that each complete labelling is also an admissible labelling. However, sometimes there exist admissible labellings that are not complete. Two examples of such labellings for the example at the left hand side of Figure 1 are $(\{B\}, \emptyset, \{A, C, D\})$ and $(\{B\}, \{C\}, \{A, D\})$.

It is interesting to notice that an admissible labelling actually corresponds with the notion of an admissible set.

Theorem 1. *Let (Ar, def) be an argumentation framework and $\mathcal{A}rgs \subseteq Ar$. $\mathcal{A}rgs$ is an admissible set iff there exists an admissible labelling $\mathcal{L}$ with* $\texttt{in}(\mathcal{L}) = \mathcal{A}rgs$*.*

The validity of Theorem 1 can be seen as follows. If $\mathcal{A}rgs$ is an admissible set, then a labelling $\mathcal{L}$ with $\texttt{in}(\mathcal{L}) = \mathcal{A}rgs$, $\texttt{out}(\mathcal{L}) = \mathcal{A}rgs^+$ and $\texttt{undec}(\mathcal{L})$ $= Ar - \texttt{in}(\mathcal{L}) - \texttt{out}(\mathcal{L})$ is an admissible labelling. Similarly, if $\mathcal{L}$ is an admissible labelling then $\texttt{in}(\mathcal{L})$ is conflict-free (otherwise at least one of the arguments in $\texttt{in}(\mathcal{L})$ would be illegally `in`). It can then be verified that $\texttt{in}(\mathcal{L})$ defends itself, due to the fact that $\mathcal{L}$ does not contain arguments that are illegally `in` or illegally `out`. We refer to [4] for a full proof.

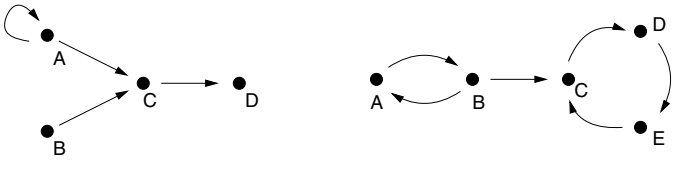


Fig. 1. Two argumentation frameworks

The notion of a complete labelling then corresponds to Dung's notion of a complete extension.

Theorem 2. *Let* (Ar, def) *be an argumentation framework and* $\mathcal{A}rgs \subseteq Ar$. $\mathcal{A}rgs$ *is a complete extension iff there exists a complete labelling* $\mathcal{L}$ *with* $\mathtt{in}(\mathcal{L}) = \mathcal{A}rgs$.

The validity of Theorem 2 can be seen as follows. If $\mathcal{A}rgs$ is a complete extension then a labelling with $\mathtt{in}(\mathcal{L}) = \mathcal{A}rgs$, $\mathtt{out}(\mathcal{L}) = \mathcal{A}rgs^+$ and $\mathtt{undec}(\mathcal{L}) = Ar - \mathtt{in}(\mathcal{L}) - \mathtt{out}(\mathcal{L})$ is a complete labelling. Similarly, if $\mathcal{L}$ is a complete labelling then $\mathtt{in}(\mathcal{L})$ is at least an admissible set (this follows from Theorem 1). It can then be verified that $\mathtt{in}(\mathcal{L})$ defends exactly itself, due to the fact that $\mathcal{L}$ does not contain any arguments that are illegally `in`, illegally `out` or illegaly `undec`. Hence, $\mathtt{in}(\mathcal{L})$ is a complete extension. Again, we refer to [4] for a full proof.

The notions of a grounded, preferred, stable and semi-stable labelling correspond to he notions of a grounded, preferred, stable and semi-stable extension, respectively.

Theorem 3. *A set* $\mathcal{A}rgs$ *of arguments is (1) a grounded extension iff there exists a grounded labelling* $\mathcal{L}$ *with* $\mathtt{in}(\mathcal{L}) = \mathcal{A}rgs$*, (2) a preferred extension iff there exists a preferred labelling* $\mathcal{L}$ *with* $\mathtt{in}(\mathcal{L}) = \mathcal{A}rgs$*, (3) a stable extension iff there exists a stable labelling* $\mathcal{L}$ *with* $\mathtt{in}(\mathcal{L}) = \mathcal{A}rgs$*, and (4) a semi-stable extension iff there exists a semi-stable labelling* $\mathcal{L}$ *with* $\mathtt{in}(\mathcal{L}) = \mathcal{A}rgs$

Before continuing with the backgrounds of the proposed algorithm, we first state a few useful properties of complete and admissible labellings.

Lemma 1. *Let* $\mathcal{L}_1$ *and* $\mathcal{L}_2$ *be two complete labellings of* (Ar, def)*. It holds that* $\mathtt{in}(\mathcal{L}_1) \subseteq \mathtt{in}(\mathcal{L}_2)$ *iff* $\mathtt{out}(\mathcal{L}_1) \subseteq \mathtt{out}(\mathcal{L}_2)$.

Lemma 2. *Let* $\mathcal{L}_1$ *be an admissible labelling. There exists a preferred labelling* $\mathcal{L}_2$ *with* $\mathtt{in}(\mathcal{L}_1) \subseteq \mathtt{in}(\mathcal{L}_2)$ *and* $\mathtt{out}(\mathcal{L}_1) \subseteq \mathtt{out}(\mathcal{L}_2)$.

Lemma 3. *Let* $\mathcal{L}$ *be a preferred labelling and* $\mathcal{L}'$ *be an admissible labelling. It holds that:*

1. *if* $\mathtt{in}(\mathcal{L}) \subseteq \mathtt{in}(\mathcal{L}')$ *then* $\mathtt{in}(\mathcal{L}) = \mathtt{in}(\mathcal{L}')$
2. *if* $\mathtt{out}(\mathcal{L}) \subseteq \mathtt{out}(\mathcal{L}')$ *then* $\mathtt{out}(\mathcal{L}) = \mathtt{out}(\mathcal{L}')$

4 Formal Background of the Algorithm

Now that the preliminary concepts have been explained, it is time to treat the main question of how to compute, given an argumentation framework, all the semi-stable labellings. The idea is to do this by generating a set $\mathcal{L}abellings$ of admissible labellings that includes at least all preferred labellings. Since every semi-stable labelling is also a preferred labelling [3,5], this means that $\mathcal{L}abellings$ also contains all semi-stable labellings. We then have to select those labellings in $\mathcal{L}abellings$ with minimal `undec` to obtain the final answer.

How does one generate an admissible labelling? A possible approach is to start with the all-in labelling (the labelling in which every argument is labelled `in`). This labelling trivially satisfies the absence of arguments that are illegally `out`. However, for an admissible labelling also the absence of arguments that are illegally `in` is required, and the all-in labelling may contain many arguments that are illegally `in`. This means we need a way of changing the label of an argument that is illegally `in`, preferrably without creating any arguments that are illegally `out`. This is done using a sequence of *transition steps*. A transition step basically takes an argument that is illegally `in` and relabels it to `out`. It then checks if, as a result of this, one or more arguments have become illegally `out`. If this is the case, then these arguments are relabelled to `undec`. More precisely, a transition step can be described as follows.

Definition 9. *Let $\mathcal{L}$ be a labelling and A an argument that is illegally* `in` *in $\mathcal{L}$. A* transition step *on A in $\mathcal{L}$ consists of the following:*

1. *the label of A is changed from* `in` *to* `out`
2. *for every $B \in \{A\} \cup A^+$, if B is illegally* `out`*, then change the label of B from* `out` *to* `undec`.

Theorem 4. *Each transition step preserves the absence of arguments that are illegally* `out`.

The validity of Theorem 4 follows directly from point 2 of Definition 9.

A transition sequence starts with an initial labelling $\mathcal{L}_0$, on which a sequence of successive transition steps is applied.

Definition 10. *A* transition sequence *is a list $[\mathcal{L}_0, A_1, \mathcal{L}_1, A_2, \mathcal{L}_2, \ldots, A_n, \mathcal{L}_n]$ ($n \geq 0$) where each A_i ($1 \leq i \leq n$) is an argument that is illegally* `in` *in labelling $\mathcal{L}_{i-1}$ and every $\mathcal{L}_i$ is the result of doing a transition step of A_i on $\mathcal{L}_{i-1}$. A transition sequence is called* terminated *iff $\mathcal{L}_n$ does not contain any argument that is illegally* `in`.

As an illustration of how a transition sequence can be constructed, consider the example at the left hand side of Figure 2. Assume the initial situation is the all-in labeling $\mathcal{L}_0 = (\{A, B, C\}, \emptyset, \emptyset)$. In this labelling both B and C are illegally `in` since each of them has a defeater that is `in`, so they are both candidates for a transition step. If we select B for a transition step, then the result is a labelling $\mathcal{L}_1 = (\{A, C\}, \{B\}, \emptyset)$. This labelling does not contain any arguments that are illegally `in`, so the transition sequence $[\mathcal{L}_0, B, \mathcal{L}_1]$ is terminated. If, at the other hand, we select C for a transition step then the result is a labelling $\mathcal{L}'_1 = (\{A, B\}, \{C\}, \emptyset)$. This labelling still has an argument that is illegally `in` (B), so we perform another transition step that relabels B from `in` to `out`. However, as a result of doing that, C becomes illegally `out` since it has no longer a defeater that is `in`, so C is relabelled from `out` to `undec`. The transition step as a whole then yields $\mathcal{L}'_2 = (\{A\}, \{B\}, \{C\})$. This means that there exists a second terminated transition sequence $[\mathcal{L}_0, C, \mathcal{L}'_1, B, \mathcal{L}'_2]$.

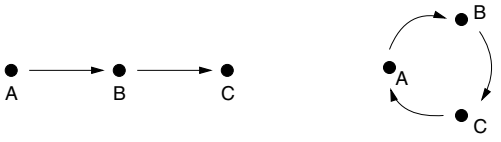


Fig. 2. Two argumentation frameworks

Now consider the example at the right hand side of Figure 2. Again, assume that the initial labelling is the all-in labelling, so $\mathcal{L}_0 = (\{A, B, C\}, \emptyset, \emptyset)$. Here, all three arguments are `in`, so each of them can be selected for a transition step. Assume, without loss of generality, that A is selected for a transition step. This then yields a labelling $\mathcal{L}_1 = (\{B, C\}, \{A\}, \emptyset)$. In this labelling, only C is illegally `in` and can be selected for a transition step. During this transition step, after C is relabelled from `in` to `out`, A becomes illegally `out` and is therefore relabelled to `undec`. Thus, the transition step as a whole yields $\mathcal{L}_2 = (\{B\}, \{C\}, \{A\})$. In this labelling B is illegally `in` since it has a defeater (A) that is `undec`. Therefore, a transition step on B is performed during which B is relabelled from `in` to `out`. Directly after doing that, however, not only C is illegally `out` but also B itself is illegally `out`, so both of them are relabelled from `out` to `undec`. Thus, the transition step as a whole yields $\mathcal{L}_3 = \{\emptyset, \emptyset, \{A, B, C\})$. This means that there exists a terminated transition sequence $[\mathcal{L}_0, A, \mathcal{L}_1, C, \mathcal{L}_2, B, \mathcal{L}_3]$. It can be verified that in the example at the right hand side of Figure 2 every terminated transition sequence that starts with the all-in labelling finishes with $\mathcal{L}_3$.

Since for any finite argumentation framework, only a finite number of successive transition steps can be performed, this means that (again for finite argumentation frameworks) each terminated transition sequence is finite. Furthermore, for any terminated transition sequence, the final labelling is an admissible labelling. This is because each transition step preserves the absence of arguments that are illegally `out` (Theorem 4) and after termination, we also do not have any arguments that are illegally `in`.

Theorem 5. *Let $[\mathcal{L}_0, A_1, \mathcal{L}_1, A_2, \mathcal{L}_2, \ldots, A_n, \mathcal{L}_n]$ ($n \geq 0$) be a terminated transition sequence where $\mathcal{L}_0$ is the all-in labelling. It holds that $\mathcal{L}_n$ is an admissible labelling.*

An interesting observation is that during the course of a transition sequence, the set of `in`-labelled arguments monotonically decreases and the set of `undec`-labelled arguments monotonically increases.

Proposition 1. *Let $[\mathcal{L}_0, A_1, \mathcal{L}_1, \ldots, A_n, \mathcal{L}_n]$ be a transition sequence. For any $1 \leq i \leq n$ it holds that $\texttt{in}(\mathcal{L}_i) \subseteq \texttt{in}(\mathcal{L}_{i-1})$ and $\texttt{undec}(\mathcal{L}_i) \supseteq \texttt{undec}(\mathcal{L}_{i-1})$.*

Proposition 1 is relevant with respect to the algorithm for generating the semi-stable labellings. Suppose that a previously generated terminated transition sequence yielded an admissible labelling $\mathcal{L}$ and we are currently expanding a transition sequence $[\mathcal{L}_0, A_1, \mathcal{L}_1, \ldots, A_i, \mathcal{L}_i]$. If $\texttt{undec}(\mathcal{L}_i) \supsetneq \texttt{undec}(\mathcal{L})$ then we already

know that the current transition sequence cannot yield a semi-stable labelling, since expanding it to a terminated transition sequence $[\mathcal{L}_0, A_1, \mathcal{L}_1, \ldots, A_n, \mathcal{L}_n]$ results in a labelling $\mathcal{L}_n$ with $\texttt{undec}(\mathcal{L}_n) \supsetneq \texttt{undec}(\mathcal{L})$ (this follows from Proposition 1 and the fact that $\texttt{undec}(\mathcal{L}_i) \supsetneq \texttt{undec}(\mathcal{L})$). We then might as well stop expanding the current transition sequence and instead try another possibility.

We define $\mathcal{L}abellings$ as the set of all final labellings from terminated transition sequences that start from the all-in labelling.

As we have now obtained that the result of any terminated transition sequence starting from the all-in labelling is an admissible labelling, it directly follows that each element of $\mathcal{L}abellings$ is an admissible labelling. The next step, then, is to examine whether each semi-stable labelling will be an element of $\mathcal{L}abellings$. If this is the case, then we can simply determine the semi-stable labellings as those elements of $\mathcal{L}abellings$ where `undec` is minimal. It turns out that this is indeed the case. This can roughly be seen as follows. Let $\mathcal{L}$ be a preferred labelling. We now construct a transition sequence that yields $\mathcal{L}$. This is done in two phases. The first phase is to perform a sequence of transition steps, starting from the all-in labelling, on each argument that is labelled `out` in $\mathcal{L}$. This yields a labelling $\mathcal{L}'$ with $\texttt{out}(\mathcal{L}') = \texttt{out}(\mathcal{L})$, $\texttt{undec}(\mathcal{L}') = \emptyset$ and $\texttt{in}(\mathcal{L}') \supseteq \texttt{in}(\mathcal{L})$. Then, during the second phase, we continue to perform transition steps, starting from $\mathcal{L}'$, until we have reached termination; that is, until there are no arguments that are illegally `in` anymore, yielding a labelling $\mathcal{L}''$. It can be verifed that this does not change he arguments that are `out` in $\mathcal{L}'$. Also, it cannot change the arguments that are `in` in $\mathcal{L}$, since these are legally `in` in $\mathcal{L}'$. That is, we have that $\texttt{out}(\mathcal{L}'') = \texttt{out}(\mathcal{L})$ and $\texttt{in}(\mathcal{L}'') \supseteq \texttt{in}(\mathcal{L})$. From the fact that $\mathcal{L}$ is a preferred labelling, it follows that (Lemma 3) $\texttt{in}(\mathcal{L}'')$ cannot be a strict superset of $\texttt{in}(\mathcal{L})$. Therefore, we have that $\texttt{in}(\mathcal{L}'') = \texttt{in}(\mathcal{L})$. From the fact that $\texttt{in}(\mathcal{L})$ and $\texttt{out}(\mathcal{L}'') = \texttt{out}(\mathcal{L})$ it follows that $\texttt{undec}(\mathcal{L}'') = \texttt{undec}(\mathcal{L})$, which implies that $\mathcal{L}'' = \mathcal{L}$. This leads to the following theorem.

Theorem 6. *Let $\mathcal{L}$ be a preferred labelling. There exists a transition sequence $[\mathcal{L}_0, A_1, \mathcal{L}_1, \ldots, A_n, \mathcal{L}_n]$ with $\mathcal{L}_0$ the all-in labelling and $\mathcal{L}_n = \mathcal{L}$.*

From the fact that each semi-stable labelling is also a preferred labelling, it then follows that for each semi-stable labelling, there exists a transition sequence that yields it.

5 Optimizing the Algorithm

As was shown in Section 4, for the example at the left hand side of Figure 2 there are two terminated transition sequences starting from the all-in labelling: one that yields $(\{A, C\}, \{B\}, \emptyset)$ and one that yields $(\{A\}, \{B\}, \{C\})$. This is because starting from the all-in labelling, we have two choices of arguments to do a transition step on: B or C, since both of them are illegally `in` in $\mathcal{L}_0$ (the all-in labelling). If we choose B we will finally end up with a complete labelling, but if we choose C then we will ultimately end up with a labelling that is admissible but not complete (and therefore also not preferred or semi-stable). An interesting question, therefore, is whether there is a way of avoiding

such non-complete results by carefully choosing the right arguments to do the transition steps on. While in general this question is difficult to answer, we do propose a simple guideline that is helpful in many cases: choose an argument that is *superillegally* `in` to do a transition step on, if such an argument is available.

Definition 11. *Let $\mathcal{L}$ be a labelling of (Ar, def). An argument A is* superillegally `in` *in $\mathcal{L}$ iff A is labelled* `in` *by $\mathcal{L}$ and is defeated by an argument that is legally* `in` *in $\mathcal{L}$ or* `undec` *in $\mathcal{L}$.*

It directly follows that if an argument is superillegally `in` in $\mathcal{L}$, then it is also illegally `in` in $\mathcal{L}$. The converse, however, may not be the case. As an example, consider again the example at the left hand side of Figure 2. With the all-in labelling, A is legally `in`, B and C are illegally `in`, and only B is superillegally `in`. Thus, it makes sense to select B to do a transition step on.

The reason why arguments that are superillegally `in` are such good candidates to perform a transition step on is that an argument that is superillegally `in` will stay illegally `in` (although it may not necessarily stay *super*illegally `in`) throughout the transition sequence, until a transition step is done on it. Thus, we might as well perform a transition step on the superillegal argument as soon as possible, since this prevents us from doing things we later regret (like performing a transition step on argument C).

Theorem 7. *Let $\mathcal{L}_0$ be a labelling where argument A is superillegally* `in` *and $[\mathcal{L}_0, A_1, \mathcal{L}_1, \ldots, A_n, \mathcal{L}_n]$ be a transaction sequence where no transaction step is performed on A (that is: $A \notin \{A_1, \ldots, A_n\}$). It holds that A is illegally* `in` *in $\mathcal{L}_n$.*

From Theorem 7 it follows that it may be a good strategy to select an argument that is superillegally `in` to do a transition step on, whenever such an argument is available. An interesting question is how such a strategy would affect the results that were obtained earlier regarding correctness (each transition sequence terminates with an admissible labelling) and completeness (for each preferred labelling, there exists a transition sequence that produces this labelling).

As for correctness, the situation does not change. The result of a terminated transition sequence is always an admissible labelling, regardless of which strategy was used to select the arguments to do transition steps on. In [4] it is explained that the new strategy also does not affect the completeness of the algorithm. That is, if we consequently choose an (arbitrary) superillegal argument to do a transition step on whenever such an argument is available, then we are still able to produce all preferred labellings, and therefore also all semi-stable labellings.

6 The Actual Algorithm

Since the algoritm starts with the labelling in which every argument is labelled `in`, we assume the presence of the constant `all_in`, which stands for the all-in labelling. There is one global variable (`pot_semi-stables`) which stands the potential semi-stable labellings, that is, the admissible labellings with minimal `undec` that have

been found until now. If, during the search algorithm, one finds that the current labelling is worse (that is: it has a proper superset of `undec` labelled arguments) than an admissible labelling found earlier, then it is time to stop evaluating the current transition sequence, since its final result will not be semi-stable anyway.

If there is no argument that is illegally `in` then we are at the end of a terminated transition sequence and have obtained an admissible labelling. From the previous check, we already know that this admissible labelling is not any worse than what we already have found (it does not have a proper superset of `undec` labelled arguments compared to a previously computed admissible labelling), so we add it to the set of potential semi-stable labellings (`pot_semi-stables`). We then have to check if we found something that is actually *better* than what we found earlier. If so, we need to delete some of the old results (remove it from `pot_semi-stables`).

If we have not reached the end of a terminated transition sequence, then there is at least one argument that is still illegally `in`. We then distinguish two cases. If there is at least one argument that is superillegally `in` then go for the argument that is superillegally `in`. There is no need to be selective; any argument that is superillegally `in` will do for a transition step. If, however, there is no argument that is superillegally `in` then we have to try each argument that is "normally" illegally `in`.

```
01. pot_semi-stables = ∅; find_semi-stables(all-in);
02. print pot_semi-stables; end;
03.
04. procedure find_semi-stables(ℒ)
05.    # if we have something worse than found earlier,
06.    # then prune the search tree and backtrack
07.    if ∃ℒ′ ∈ pot_semi-stables: undec(ℒ′) ⊊ undec(ℒ) then return;
08.    # now see if the transition sequence has terminated
09.    if ℒ does not have an argument that is illegally in then
10.      for each ℒ′ ∈ pot_semi-stables
11.        # if old result is worse than new labelling: remove
12.        if undec(ℒ) ⊊ undec(ℒ′) then
13.          pot_semi-stables := pot_semi-stables - ℒ′;
14.        endif;
15.      endfor;
16.     # add our newly found labelling as a candidate; we already
17.     # know that it is not worse than what we already have
18.      pot_semi-stables := pot_semi-stables ∪ ℒ;
19.      return; # we are done with this one; try next possibility
20.    else
21.      if ℒ has an argument that is superillegally in then
22.        A := some argument that is superillegally in in ℒ;
23.        find_semi-stables(transition_step(A, ℒ));
24.      else
25.        for each argument A that is illegally in in ℒ
```

```
26.          find_semi-stables(transition_step(A, L));
27.        endfor;
28.      endif;
29.    endif;
30. endproc;
```

7 Discussion

It is interesting to observe that the algorithm stated in Section 6 can also be used to calculate, respectively, stable semantics and preferred semantics, by applying a few changes.

For stable semantics, the modification is quite straightforward. Basically, the idea (Definition 8) is only to yield labellings without `undec`-labelled arguments. For this, we have to stop expanding a transition sequence as soon as an `undec`-labelled argument is produced. Therefore, we have to replace line 7 by `if undec`$(\mathcal{L}) \neq \emptyset$ `then return`.
Furthermore, we do not have to compare the sets of `undec`-labelled arguments of the previous results with the current result, so the lines 10 until 15 can be removed. Then, after renaming the variable `pot_semi-stables` to `stables` and renaming the procedure `find_semi-stables` to `find_stables`, the modifications are finished and the result is an algorithm that calculates all stable extensions of an argumentation framework.

For preferred semantics, the modification is slightly different. The idea is that we still have to check for a condition that allows us to cut off the current transition sequence once we know that it will not yield a useful result. For semi-stable semantics, it can be observed that the set of `undec`-labelled arguments keeps *increasing* as the transition sequence progresses (Proposition 1). For preferred semantics, it can be observed that the set of `in`-labelled arguments keeps *decreasing* as the transition sequence progresses (Proposition 1). In both cases, there may come a point where the current transition sequence becomes worse than a result found earlier, which means we might as well stop expanding it and instead backtrack to another possibility.

The modification for preferred semantics is done as follows. First, the variable `pot_semi-stables` to renamed as `pot_preferreds` and the procedure `find_semi-stables` is renamed as `find_preferreds`. Line 7 is replaced by: `if` $\exists\mathcal{L}' \in$ `pot_preferreds: in`$(\mathcal{L}') \supsetneq$ `in`$(\mathcal{L})$ `then return;`
Line 12 is replaced by: `if in`$(\mathcal{L}) \supsetneq$ `in`$(\mathcal{L}')$ `then`
The result, then, is an algorithm that calculates, given an argumentation framework, all preferred extensions.

In [5], it was first examined how argument labellings are related to the traditional Dung-style argument semantics.It was found that a complete labelling has a maximal set of `in`-labelled arguments iff it has a maximal set of `out`-labelled arguments. In both cases, the labelling corresponds with a preferred extension. Furthermore, it was found that a complete labelling has a minimal set of `in`-labelled arguments iff it has a minimal set of `out`-labeled arguments iff

it has a maximal set of `undec`-labelled arguments. In all three cases, the labelling corresponds with the grounded extension. The only option left to be examined consists of the labellings where the set of `undec`-labelled arguments is minimal. It turned out that these did not correspond with any well-known semantics, and this is how semi-stable semantics was discovered [3,5]. Thus, one can perhaps think of semi-stable semantics as a missing link in the traditional hierarchy of argumentation semantics.

Nevertheless, semi-stable semantics is more than just a purely theoretical notion. Stable semantics, despite its property of the potential absence of stable extensions, is still being used for the purpose of constraint satisfaction in fields like answer set programming [7]. The idea is that a problem is specified in a declarative way and that the set of potential solutions then corresponds with the stable models of the thus described problem. In cases where no solutions exists, there should therefore also not exist any stable models. Thus, the absence of stable models (or extensions) is not always an undesirable property. This does assume, however, that the original problem was encoded in a way that is perfectly correct. If, for instance, an answer set program contains an error, then the result may well be a total absence of stable models, which is a situation that can be notoriously hard to debug. With semi-stable semantics, however, one obtains one or more models that can serve as a starting point to examine where things went wrong. For instance, consider the example at the left hand side of Figure 1. It has no stable extensions and its (only) semi-stable extension is $(\{B, D\}, \{C\}, \{A\})$. The fact that A is labelled `undec` can be seen as an indication of what is "wrong" in this argumentation framework from the perspective of stable semantics. Similarly, if there exists an odd loop that causes the absence of stable models, then this odd loop is flagged `undec` by a semi-stable model. Thus, semi-stable semantics can give a good indication of where to start debugging if no stable model exists. Semi-stable semantics does a better job here than, for instance, preferred semantics. This is because there can be non-stable preferred models (like $(\{A\}, \{B\}, \{C, D, E\})$) even in cases where stable models (like $(\{B, D\}, \{A, C, E\}, \emptyset)$ do exist. With semi-stable semantics, one obtains non-stable semi-stable models only if there is a real problem that prevents the existence of stable models.

One particularly interesting application of semi-stable semantics would be answer set programming and other forms of logic programming that use the stable model semantics. At the time of writing, the author is exploring the possibilities of applying semi-stable semantics to logic programming and answer set programming. We believe this would be a useful approach for analyzing programs for which no stable models exist.

References

1. Vreeswijk, G.: An algorithm to compute minimally grounded and admissible defence sets in argument systems. In: Dunne, P., Bench-Capon, T. (eds.) Computational Models of Argument; Proceedings of COMMA 2006, pp. 109–120. IOS, Amsterdam (2006)

2. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77, 321–357 (1995)
3. Caminada, M.: Semi-stable semantics. In: Dunne, P., Bench-Capon, T. (eds.) Computational Models of Argument; Proceedings of COMMA 2006, pp. 121–130. IOS Press, Amsterdam (2006)
4. Caminada, M.: An algorithm for computing semi-stable semantics. Technical Report Technical Report UU-CS-2007 -010, Utrecht University (2007), http://www.cs.uu.nl/~martinc/algorithm_techreport.pdf
5. Caminada, M.: On the issue of reinstatement in argumentation. In: Fisher, M., van der Hoek, W., Konev, B., Lisitsa, A. (eds.) JELIA 2006. LNCS (LNAI), vol. 4160, pp. 111–123. Springer, Heidelberg (2006)
6. Pollock, J.L.: Cognitive Carpentry. In: A Blueprint for How to Build a Person, MIT Press, Cambridge (1995)
7. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Computing 9(3/4), 365–385 (1991)

The Logical Handling of Threats, Rewards, Tips, and Warnings

Leila Amgoud, Jean-Francois Bonnefon, and Henri Prade

Institut de Recherche en Informatique de Toulouse (IRIT)
118, route de Narbonne,
31062 Toulouse Cedex 4 France
{amgoud, bonnefon, prade}@irit.fr

Abstract. Previous logic-based handling of arguments has mainly focused on explanation or justification in presence of inconsistency. As a consequence, only one type of argument has been considered, namely the explanatory type; several argumentation frameworks have been proposed for generating and evaluating explanatory arguments. However, recent investigations of argument-based negotiation have emphasized other types of arguments, such as *threats*, *rewards*, *tips*, and *warnings*. In parallel, cognitive psychologists recently started studying the characteristics of these different types of arguments, and the conditions under which they have their desired effect. Bringing together these two lines of research, we present in this article some logical definitions as well as some criteria for evaluating each type of argument. Empirical findings from cognitive psychology validate these formal results.

Keywords: Argumentation, Negotiation, Threats/Rewards, Tips/Warn ings.

1 Introduction

Argumentation is an established approach for reasoning with inconsistent knowledge, based on the construction and the comparison of arguments, and it may also be considered as an alternative method for handling qualitative uncertainty. A basic idea behind argumentation is that it should be possible to say more about the certainty of a particular fact than just assessing a certainty degree in $[0, 1]$. In particular, it should be possible to assess the reason why a fact holds, under the form of arguments, and combine these arguments for evaluating the certainty of the fact they support. This combination process can be viewed as determining the most acceptable among arguments.

Various argument-based frameworks have been developed in defeasible reasoning [1,6,8,20,22], for generating as well as for evaluating arguments. However, in that explanation-oriented perspective, only one type of argument has been considered, namely the *explanatory* type (reasons for believing, explanations for states of affairs). Yet, another line of work [2,15,19] suggests that argumentation can also play a key role in negotiation: E.g., an offer supported by a good argument has a better chance of being accepted by another agent. Argumentation

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 235–246, 2007.

may also lead an agent to change its goals, or may impose a particular response onto an agent.

In addition to the explanatory arguments studied in classical argumentation frameworks, the above works have emphasized other types of arguments such as inducements, deterrents, and pieces of advice. For example, if an agent receives a *threat*, it may accept an offer even though this offer has no particular appeal, so as not to jeopardize the truly important goals targeted by the threat. In parallel, cognitive psychologists have studied in recent years the characteristics of these different types of arguments, and the conditions under which they have their desired effect. Bringing together these lines of research, we present in this article the formal definitions of the four basic non-explanatory arguments (threats, rewards/promises, warnings, and tips[1]), as well as some criteria for evaluating them. Empirical findings from cognitive psychology validate our formal results.

2 The Four Basic Non-explanatory Arguments

It has been pointed out that it is not possible to present an exhaustive classification of arguments, because arguments operate within a particular context and domain [24]. For example, when inferring from inconsistent knowledge bases, arguments aim at finding the most supported beliefs. But during a negotiation, the exchange of arguments may lead the agent which receives them to change its goals or preferences.

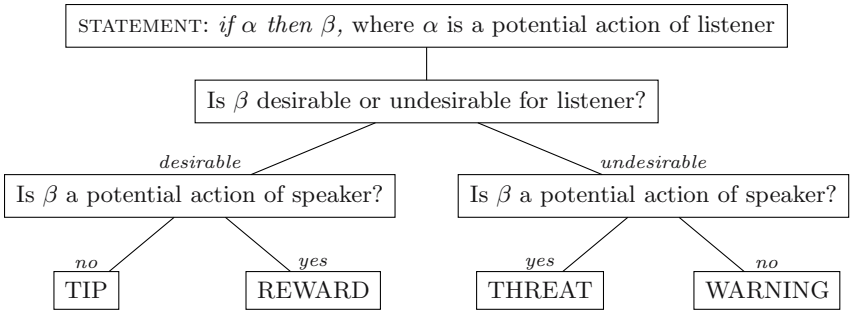


Fig. 1. A decision tree for classifying arguments, adapted from [17]

Nonetheless, some typologies exist that consider the kinds of arguments thought to have persuasive force in human negotiation, both in artificial intelligence [15] and in cognitive psychology [17]. Building on previous research [9,16], López-Rousseau and Ketelaar recently tested a simple yet remarkably efficient algorithm for predicting whether people will think of a given conditional statement as expressing a threat, a reward, a tip, or a warning (see Figure 1).

[1] Although the term 'tip' can evoke a small piece of heuristic information for making something better, it must be understood here in the sense of a *recommendation*.

Consider that a speaker is telling a listener: "If you do α, β will happen." (Note that α is necessarily a potential action of the listener.) The algorithm of [17] focuses on two characteristics of β, namely: Is β something the speaker will do, or something that will happen independently of the speaker? Is β something good for the listener, or something bad? Quite remarkably, this simple algorithm correctly predicted 92% of the classifications made by human subjects.

In parallel, other authors [4] elaborated an in-depth psychological analysis of the motivational structure of such statements, and emphasized that the action α itself, inasmuch as rewards or threats are concerned, should have positive or negative consequences for the speaker. Indeed, why would the speaker attempt to bribe the listener into doing α if the speaker had no interest in seeing that α is done? Likewise, why would the speaker attempt to scare the listener out of doing α if the speaker had no interest in seeing that α is not done?

Our goal in this article is to organize psychological analysis and empirical results in a formal framework that will do justice to the psychological state of the art. To our knowledge, this framework is the first to address all four types of non-explanatory arguments, as well as the first to be entirely grounded in experimental research.

3 Formal Definitions

In what follows, $\mathcal{AC}$ denotes a set of actions. $\{a_1, \ldots, a_n\}$ is a set of agents involved in a discussion. In addition to this set of agents, we suppose that we have a neutral agent, denoted by a_0, that may stand for impersonal powers such as Nature itself. Let $\mathcal{AG} = \{a_0, a_1, \ldots, a_n\}$ be the set of all agents. Each agent is supposed to have the control over a subset of actions of $\mathcal{AC}$. This captures the fact that an agent is able to do some actions but not others. The function

$$\mathcal{F} : \mathcal{AG} \longrightarrow 2^{\mathcal{AC}}$$

retuns the actions under the control of each agent.

Each action performed by a given agent (including the neutral agent) is supposed to have consequences for all agents. These consequences can be good, neutral, or bad, and can be good or bad to different degrees. This notion of consequence is captured by the following function:

$$\texttt{Cons} : \mathcal{AC} \times \mathcal{AG} \longrightarrow \{-n, \ldots, 0, \ldots, +n\},$$

where n is an integer that denotes the extremity of the consequence of some action to some agent. Positive values of `Cons` denote good consequences, the higher the value of `Cons` the better. Negative values of `Cons` denote bad consequences, the lower the value of `Cons` the worse. The value 0 is attached to neutral consequences. This simple, ordinal scale can be generalized to more sophisticated scales, providing that they include a neutral point. Throughout the paper, we suppose that agent S, the speaker, addresses a negotiation move (a statement) to agent L, the listener, with $S, L \in \{a_1, \ldots, a_n\}$.

Definition 1 (Argument). *An argument is an expression of the form* $(a_i, \alpha) \longrightarrow (a_j, \beta)$ *such that:*

1. $a_i, a_j \in \mathcal{AG}$,
2. $\alpha, \beta \in \mathcal{AC}$

The meaning of the above expression is that if agent a_i performs action α, the agent a_j will perform action β.

3.1 Threats

Threats are used to coerce an agent into behaving in a certain way, by emphasizing the unpleasant measures the speaker would take otherwise. Different linguistic expressions of threats are possible. The conditional expression is canonical, but threats can easily be reformulated as conjunctions or disjunctions [11].

i) If you do α, I will do β,
ii) Do α and I will do β,
iii) Do not do α otherwise I will do β.

Example 1 (Tantrum).

i) If you throw a tantrum, I'll ground you.
ii) Throw a tantrum and I will ground you.
iii) Don't throw a tantrum, otherwise I will ground you.

Definition 2 (Threat). *An argument of type threat, or a* threat*, is an argument* $(a_i, \alpha) \longrightarrow (a_j, \beta)$ *such that:*

1. $a_i = L$
2. $\alpha \in \mathcal{F}(a_i)$
3. $a_j = S$
4. $\texttt{Cons}(\alpha, a_j) < 0$
5. $\texttt{Cons}(\beta, a_i) < 0$

Since $S, L \in \{a_1, \ldots, a_n\}$, neither can be the neutral agent. Points 1 and 2 are common to the definition of threats, rewards, tips, and warnings. They ensure that α is an action under the control of the listener; otherwise, the threat (reward, etc.) would be *useless*. Point 3 ensures that β is an action of the speaker, a characteristic feature of threats and warnings.[2] Point 4 ensures that the speaker does not attempt to prevent something that would actually be beneficial; otherwise, the threat would be *irrational*. Finally, Point 5 ensures that β is something unpleasant to the listener; otherwise, the threat would be *misplaced.*

In Example 1, α is meant to be an action of the listener (a child), namely, throwing a tantrum. This action (or lack thereof) is presumed to be under the control of the child. In contrast, β is meant to be an action of the speaker (the mother), namely, grounding the child. The child throwing a tantrum is something unpleasant to the mother, and being grounded is something unpleasant to the child. The statement meets all the criteria in the definition of a threat.

[2] We will return in section 5 to the fact that Definition 2 does not feature the condition $\beta \in \mathcal{F}(a_j)$.

3.2 Rewards

Rewards are used to encourage another agent to behave in a certain way, by emphasizing the pleasant measures the speaker will take in response. There are two main linguistic expressions of rewards. As for threats, the conditional expression of rewards is canonical, but the conjunctive reformulation is possible. Unlike threats, the disjunctive paraphrase is awkward [11].

i) If you do α, I will do β.
ii) Do α and I will do β.

Example 2 (Free CDS).

i) If you buy this computer, I'll throw in a box of free CDs.
ii) Buy this computer and I'll throw in a box of free CDs.
iii) Don't buy this computer, otherwise I'll throw in a box of free CDs.

Definition 3 (Reward). *An argument of type reward, or a* reward*, is an argument* $(a_i, \alpha) \longrightarrow (a_j, \beta)$ *such that:*

1. $a_i = L$
2. $\alpha \in \mathcal{F}(a_i)$
3. $a_j = S$
4. $\texttt{Cons}(\alpha, a_j) > 0$
5. $\texttt{Cons}(\beta, a_i) > 0$

Note that, due to the fact that $S, L \in \{a_1, \ldots, a_n\}$, neither can be the neutral agent. Points 1 and 2 serve the same function as in the definition of threats. Point 3 ensures that β is meant to be an action of the speaker, a characteristic feature of threats and rewards. Point 4 ensures that the speaker does not attempt to bring about something that would actually be detrimental; otherwise, the reward would be *irrational*. Finally, Point 5 ensures that β is something pleasant to the listener; otherwise, the reward would be *misplaced.*

In Example 2, α is an action under the control of the listener of the listener (a customer), namely, buying a computer. In contrast, β is an action of the speaker (the salesperson), namely, throwing in a box of free CDs. The customer buying a computer is something desirable for the salesperson, and being given a box of free CDs is something desirable to the customer. The statement meets all the criteria in the definition of a reward.

3.3 Warnings

Warnings are addressed to another agent in an attempt to discourage a given course of action, by emphasizing the unfortunate consequences that would follow. In contrast to threats, these unfortunate consequences are not within the control of the speaker [10], and the speaker has no particular stake in preventing the course of action to occur [18]. Just as threats, warnings can be formulated conditionally, conjunctively, or disjunctively.

Example 3 (Computer Virus).

i) If you open this file, your computer will crash.
ii) Open this file and your computer will crash.
iii) Don't open this file, otherwise your computer will crash.

Definition 4 (Warning). *An argument of type warning, or a* warning*, is an argument* $(a_i, \alpha) \longrightarrow (a_j, \beta)$ *such that:*

1. $a_i = L$
2. $\alpha \in \mathcal{F}(a_i)$
3. $a_j \neq S$
4. $\texttt{Cons}(\beta, a_i) < 0$

Points 1 and 2 are common to all four definitions. Point 3 states β should be an action of another agent than the speaker (possibly an action by the impersonal agent). This is characteristic of tips and warnings. Point 4 ensures that β is an unpleasant consequence for the listener; otherwise, the warning would be *misplaced.* Note that the definition of a warning differs in two important respects from that of a threat. First, β is not an action of the speaker; second, it is not necessary (though not excluded, either) that α harms the speaker.

In Example 3, α is meant to be an action of the listener (a computer user), namely, opening a file. This action is under the control of the user. Action β (namely, crashing the computer) is not meant to be an action of the speaker (some hotline operator), but an action of a 'neutral' agent, the computer virus. Finally, whilst the crashing of the computer is certainly undesirable to the listener, the opening of the file is of no concern to the speaker. The statement meets all the criteria in the definition of a warning, but not the criteria in the definition of a threat (or a reward, of course).

3.4 Tips

Tips are addressed to another agent in an attempt to encourage a given course of action, by emphasizing the positive consequences that would follow. In contrast with rewards, these positive consequences are not within the control of the speaker, and the speaker has no particular stake in seeing that the course of action is taken. Just as rewards, tips can be formulated conditionally or conjunctively, but sound awkward when paraphrased disjunctively.

Example 4 (Revise and Resubmit).

i) If you revise the paper, the editor will accept it.
ii) Revise the paper and the editor will accept it.
iii) Don't revise the paper, otherwise the editor will accept it.

Definition 5 (Tip). *An argument of type tip, or a* tip*, is an argument* $(a_i, \alpha) \longrightarrow (a_j, \beta)$ *such that:*

1. $a_i = L$
2. $\alpha \in \mathcal{F}(a_i)$
3. $a_j \neq S$
4. $\texttt{Cons}(\beta, a_i) > 0$

Points 1 and 2 are common to all four definitions. Point 3 states β should be an action of another agent than the speaker (possibly an action by the impersonal agent). This is characteristic of tips and warnings. Point 5 ensures that β is indeed a pleasant consequence for the listener; otherwise, the tip would be *misplaced.* Note that the definition of a tip differs from that of a reward in two different respects. First, β is not an action of the speaker; second, it is not necessary (though not excluded, either) that α benefits the speaker.

In Example 4, α is meant to be an action of the listener (a graduate student), namely, revising a paper. This action is under the control of the student. Action β (namely, accepting the paper) is not meant to be an action of the speaker (a post-doctoral student met at a conference), but an action of a third agent, the editor. Finally, the acceptance of the paper is of course desirable to the listener, but the speaker has no particular stake in seeing that the paper is revised. The statement meets all the criteria in the definition of a tip, but not the criteria in the definition of a reward (or a threat, or a warning).

4 General Properties

We assume *symmetrical control:* An agent who controls action α also controls $\neg\alpha$: $\alpha \in \mathcal{F}(a_i) \iff \neg\alpha \in \mathcal{F}(a_i)$.[3] Furthermore, we assume *bipolar consequences:* When an action α has positive consequences for an agent, $\neg\alpha$ has negative consequences for this same agent: $\texttt{Cons}(\alpha, a_i) > 0 \iff \texttt{Cons}(\neg\alpha, a_i) < 0$.

Proposition 1 (Exclusive Definitions). *An argument* $(a_i, \alpha) \longrightarrow (a_j, \beta)$ *is either a threat, or a reward, or a tip, or a warning, or none of these. All* ors *in the previous sentence are exclusive.*

It follows trivially from the definitions we have given that an argument can only meet the criteria in one definition, but not two. An example of an argument that does not satisfy any of the four definitions is $(a_i, \alpha) \longrightarrow (a_0, \beta)$, where $a_i \in \mathcal{AG} \setminus \{S, L\}$. E.g., 'If my CEO admits the fraud, her stocks will go down.' We will get back to this kind of 'consequential arguments' [7] in the final section of this article. Although Proposition 1 is straightforward, it is a genuine improvement over previous frameworks that failed to give non-overlapping definitions of threats, rewards, tips, and warnings [3,12].

Proposition 2 (From Threats to Rewards). *If* $(a_i, \alpha) \longrightarrow (a_j, \beta)$ *is a threat, then* $(a_i, \neg\alpha) \longrightarrow (a_j, \gamma)$ *is a reward for any* γ *such that* $\texttt{Cons}(\gamma, a_j) > 0$.

[3] Note that $\neg\alpha$ means 'not executing α' and not 'executing some action with the complementary effect of α'.

Proof. If $(a_i, \alpha) \longrightarrow (a_j, \beta)$ is a threat, then $a_i = L$, $\alpha \in \mathcal{F}(a_i)$, and $a_j = S$. The first three criteria for $(a_i, \neg\alpha) \longrightarrow (a_j, \gamma)$ to be a reward are thus satisfied. Furthermore, $\texttt{Cons}(\alpha, a_i) < 0$, which implies, under the assumption we have made, that $\texttt{Cons}(\neg\alpha, a_i) > 0$. The fourth criteria is satisfied. What remains to be satisfied is the fifth criteria, i.e., $\texttt{Cons}(\gamma, a_j) > 0$. Note that this criterion will be automatically satisfied in the particular case where γ is $\neg\beta$.

Proposition 3 (From Rewards to Threats). *If* $(a_i, \alpha) \longrightarrow (a_j, \beta)$ *is a reward, then* $(a_i, \neg\alpha) \longrightarrow (a_j, \gamma)$ *is a threat for any* γ *such that* $\texttt{Cons}(\gamma, a_j) < 0$.

Proof. Proof is similar to that of Proposition 2, and the same remark holds about the particular case where γ is $\neg\beta$.

Example 5 (Threat to Reward, and Vice Versa). *The threat 'If you throw a tantrum, I'll ground you' becomes a reward when its antecedent is negated and its consequent replaced by anything desirable to the listener, e.g., 'If you don't throw a tantrum, we'll come back here another time.' The reward 'If you buy this computer, I'll throw in a box of free CDs' becomes a threat when its antecedent is negated and its consequent replaced by anything undesirable to the listener, e.g., 'If you don't buy this computer, I'll tell your wife about our affair.'*

Proposition 4 (From Warnings to Tips). *If* $(a_i, \alpha) \longrightarrow (a_j, \beta)$ *is a warning, then* $(a_i, \neg\alpha) \longrightarrow (a_j, \gamma)$ *is a tip for any* γ *such that* $\texttt{Cons}(\gamma, a_j) > 0$.

Proof. If $(a_i, \alpha) \longrightarrow (a_j, \beta)$ is a warning, then $a_i = L$, $\alpha \in \mathcal{F}(a_i)$, and $a_j \neq S$. The first three criteria for $(a_i, \neg\alpha) \longrightarrow (a_j, \gamma)$ to be a tip are thus satisfied. What remains to be satisfied is the fourth criteria, i.e., $\texttt{Cons}(\gamma, a_j) > 0$. Note that this criterion will be automatically satisfied in the particular case where γ is $\neg\beta$.

Proposition 5 (From Tips to Warnings). *If* $(a_i, \alpha) \longrightarrow (a_j, \beta)$ *is a tip, then* $(a_i, \neg\alpha) \longrightarrow (a_j, \gamma)$ *is a warning for any* γ *such that* $\texttt{Cons}(\gamma, a_j) < 0$.

Proof. Proof is similar to that of Proposition 4, and the same remark holds about the particular case where γ is $\neg\beta$.

Example 6 (Warning to Tip, and Vice Versa). *The warning 'If you open this file, your computer will crash' becomes a tip when its antecedent is negated and its consequent replaced by anything desirable to the listener, e.g., 'If you don't open this file, you can claim you never received it.' The tip 'If you revise the paper, the editor will accept it' becomes a warning when its antecedent is negated and its consequent replaced by anything undesirable to the listener, e.g., 'If you don't revise the paper, your co-authors will think poorly of you.'*

5 The Strength of Non-explanatory Arguments

It is a standard perspective in argumentation research to assume that arguments differ in strength, or persuasive force. This makes it possible for an agent to

compare arguments and select the strongest one. In [3], different definitions are proposed for computing the strength of threats and rewards. These computations are based on the quality of information used to build the arguments. Within this framework, threats and rewards are built from a knowledge base and a base of goals. Thus, the strength of a threat will depend on the *certainty level* of the beliefs used to build that threat, and on the *importance* of the threatened goal. A threat is strong if it invalidates an important goal according to the most certain beliefs. A threat is weaker if it involves beliefs of low certainty, or if it only invalidates a goal of low importance. This framework, however, does not entirely do justice to the complexity of evaluating threats. Even if a threat does target an important goal of the listener and involves highly certain beliefs, other aspects of the situation can make it weak, as suggested by experimental results available in the cognitive psychology literature. Formally:

Definition 6 (Force of a Threat).
A threat $(a_i, \alpha) \longrightarrow (a_j, \beta)$ *is* strong *iff:*

- $\beta \in \mathcal{F}(a_j)$, *and*
- $\texttt{Cons}(\beta, a_j) \geq 0$, *and*
- $|\texttt{Cons}(\beta, a_i)|$ - $|\texttt{Cons}(\alpha, a_j)| \leq \delta$, *where* δ *is a threshold.*

Otherwise, the threat is weak.

The first condition says that the action β should be under the control of the speaker. If it is not, the threat is 'degenerated' [4], and will have little effect on the listener, as empirically shown in, e.g., [18]. One might try to threaten a journal editor to commit her to a psychiatric ward if one's paper is not accepted, but such a threat is unlikely to be taken seriously, as the speaker is unlikely to have such a power. The second condition says that a threat is stronger if action β has a positive side effect for the speaker, or, at least, does not harm the speaker. The sentence 'If you don't behave, we will leave immediately' has more weight if the speaker is a mother looking forward to going home, than if she is a mother who took her child to an important meeting. The third condition is more subtle. It says that the threat should not be disproportionate, i.e., that the punishment should be balanced to the offense if the threat is to be taken seriously. As shown empirically by [25], a proportionate threat such as 'If you tell your brother that Santa does not exist, I'll ground you' is much more efficient than its disproportionate version 'If you tell your brother that Santa does not exist, we will return all your presents to the store.' Remarkably, this result does not hold for rewards, as reflected in the following definition.

Definition 7 (Force of a Reward).
A reward $(a_i, \alpha) \longrightarrow (a_j, \beta)$ *is* strong *iff:*

- $\beta \in \mathcal{F}(a_j)$, *and*
- $\texttt{Cons}(\beta, a_j) \geq 0$.

Otherwise, the reward is weak.

Just like threats, a reward is strong only if it is indeed in the power of the speaker to deliver the reward β. The reward is even more convincing if the speaker finds a positive side effect in doing β. Unlike threats, rewards do not have to be proportionate, because unlike threats, rewards engage the speaker [4,12]. As shown in [25], the reward 'If you behave, I'll give you $100' is just as credible that the reward 'If you behave, I'll let you watch a cartoon tonight.' While individuals may think that promising $100 to a child is not very good parenting, they do not question the fact that the parent will stay true to her promise and deliver the $100 if the child does behave. These preliminary results do not preclude the possibility that a limit may exist beyond which a reward is no longer credible (as a function of the speaker's resources? as a function of the listener's assumptions about what a fair reward should be?). Maybe this limit is only much more flexible for rewards than it is for threats—this is still, however, an open empirical question.

Tips and warnings do not seem to have special requirements to be strong. However, as for threats and rewards, a necessary condition for a tip or a warning to be strong is that it is indeed in the power of the third (possibly neutral) agent a_j to take action β. Note that a tip (resp., a warning) might seem even stronger is $\texttt{Cons}(\alpha, a_j) < 0$ (resp., $\texttt{Cons}(\alpha, a_j) > 0$)—that is, when the speaker suggests a course of action that would be beneficial to the listener but detrimental to the speaker herself, or when the speaker warns against a course of action that would be detrimental to the listener but beneficial to the speaker herself. We do not know, however, of any experimental data that would back up this intuition.

6 Related Works

This article does not deal with the notion of threat that is pervasive in research on decision under risk, nor with the notion of threat that is involved in engineering applications such as military target analysis, or intrusion detection in computer security. Such applications revolve around evaluating how certain the threat is, and how important its potential consequences. The use of fuzzy logic-based techniques has been proposed for both applications [5,13,14]. Rather, in this article we are concerned by the expression of a threat as a *special type of argument,* and how it is perceived by another agent. We are also interested in the dual notion of reward, and in the other duality represented by tips and warnings. For that purpose, we have proposed a formal and abstract framework, grounded in experimental results, in which these four types of arguments are defined, distinguished, and evaluated.

A relevant line of work in artificial intelligence can be found in [12,15,21], although the present approach is substantially different. [15,21] in particular do not study tips and warnings, and do not consider threats and rewards as arguments. Rather, threats and rewards are considered persuasive particles, *speech acts* having preconditions and post-conditions. The preconditions must be satisfied before sending a particle, and the post-conditions represent the consequences of that particle (more precisely, these consequences amounts to adding new

beliefs in the listener's beliefs base). A final and important difference between the two approaches is our reliance on empirical data to ground our definitions and validate our assumptions.

Cognitive psychologists have also explored arguments that are kindred to threats, rewards, tips and warnings, and that our framework should easily handle. For example, [7] showed that a statement such as 'If my CEO admits the fraud, her stocks will go down' is perceived as an argument that the CEO will not admit the fraud. The informal definition given for these 'consequential' conditionals can easily be translated into our formal framework as $(a_i, \alpha) \longrightarrow (a_0, \beta)$, where $a_i \in \mathcal{AG} | \{S, L\}$ and $\texttt{Cons}(\beta, a_i) < 0$. Likewise, the 'persuasion' conditionals studied by [23] (e.g., 'If the Kyoto accord is ratified, greenhouse gas emissions will be reduced') can easily be defined in our formal framework.

7 Conclusion

Different types of arguments are exchanged in negotiation dialogues in addition to explanatory arguments. The most common are threats, rewards, warnings and tips. Although there have been attempts at formalizing threats and rewards, no effort has been done at providing a systematic formalization of all four arguments, as well as the criteria to evaluate their strength. We have proposed such a formalization, in which the differences between these arguments are clearly identified, and their persuasive forces are discussed. Furthermore, in a collaborative effort between psychologists and computer scientists, our formal choices have been systematically guided by recent empirical findings from cognitive psychology [4,9,10,16,17,18,25]. As a result, our formalization captures exactly what we know of the way human agents exchange threats, rewards, tips, and warnings.

References

1. Amgoud, L., Cayrol, C.: A reasoning model based on the production of acceptable arguments. Annals of Mathematics and Artificial Intelligence 34, 197–216 (2002)
2. Amgoud, L., Parsons, S., Maudet, N.: Arguments, dialogue, and negotiation. In: Proceedings of the 14th European Conference on Artificial Intelligence (2000)
3. Amgoud, L., Prade, H.: Handling threats, rewards and explanatory arguments in a unified setting. International Journal Of Intelligent Systems 20, 1195–1218 (2005)
4. Beller, S., Bender, A., Kuhnmünch, G.: Understanding conditional promises and threats. Thinking and Reasoning 11, 209–238 (2005)
5. Berrached, A., Beheshti, M., de Korvin, A., Aló, R.: Applying fuzzy relation equations to threat analysis. In: Proc. 35th Annual Hawaii International Conference on System Sciences, vol. 2, pp. 50–54 (2002)
6. Besnard, P., Hunter, A.: A logic-based theory of deductive arguments. Artificial Intelligence 128, 203–235 (2001)
7. Bonnefon, J.F., Hilton, D.J.: Consequential conditionals: Invited and suppressed inferences from valued outcomes. Journal of Experimental Psychology: Learning, Memory, and Cognition 30, 28–37 (2004)

8. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77, 321–357 (1995)
9. Evans, J.S.B.T.: The social and communicative function of conditional statements. Mind & Society 4, 97–113 (2005)
10. Evans, J.S.B.T., Twyman-Musgrove, J.: Conditional reasoning with inducements and advice. Cognition 69, B11–B16 (1998)
11. Fillenbaum, S.: How to do some things with IF. In: Cotton, J.W., Klatzky, R.L. (eds.) Semantic factors in cognition, Hillsdale, NJ, pp. 169–231. Erlbaum, Mahwah (1978)
12. Guerini, M., Castelfranchi, C.: Promises and threats in persuasion. In: Proceedings of the 6th Workshop on Computational Models of Natural Argument (2006)
13. Hamed, E., Graham, J., Elmaghraby, A.: Computer system threat evaluation. In: Proc. 10th International Conference on Intelligent Systems, Washington, pp. 23–26. International Society for Computers and Their Applications, Raleigh (2001)
14. Hamed, E., Graham, J., Elmaghraby, A.: Fuzzy threat evaluation in computer security. In: Proc. International Conference on Computers and Their Applications, San Francisco, pp. 389–393. International Society for Computers and Their Applications, Raleigh (2002)
15. Kraus, S., Sycara, K., Evenchik, A.: Reaching agreements through argumentation: a logical model and implementation. Journal of Artificial Intelligence 104, 1–69 (1998)
16. López-Rousseau, A., Ketelaar, T.: "If..": Satisficing algorithms for mapping conditional statements onto social domains. European Journal of Cognitive Psychology 16, 807–823 (2004)
17. López-Rousseau, A., Ketelaar, T.: Juliet: If they do see thee, they will murder thee: A satisficing algorithm for pragmatic conditionals. Mind & Society 5, 71–77 (2006)
18. Ohm, E., Thompson, V.: Everyday reasoning with inducements and advice. Thinking and Reasoning 10, 241–272 (2004)
19. Parsons, S., Sierra, C., Jennings, N.R.: Agents that reason and negotiate by arguing. Journal of Logic and Computation 8(3), 261–292 (1998)
20. Pollock, J.L.: How to reason defeasibly. Journal of Artificial Intelligence 57, 1–42 (1992)
21. Ramchurn, S.D., Jennings, N., Sierra, C.: Persuasive negotiation for autonomous agents: a rhetorical approach. In: IJCAI Workshop on Computational Models of Natural Arguments (2003)
22. Simari, G., Loui, R.: A mathematical treatment of defeasible reasoning and its implementation. Journal of Artificial Intelligence 53, 125–157 (1992)
23. Thompson, V.A., Evans, J.S.B.T., Handley, S.J.: Persuading and dissuading by conditional argument. Journal of Memory and Language 53, 238–257 (2005)
24. Toulmin, S., Reike, R., Janik, A.: An introduction to reasoning. Macmillan Publishing Company, Inc. (1979)
25. Verbrugge, S., Dieussaert, K., Schaeken, W., Van Belle, W.: Promise is debt, threat another matter: The effect of credibility on the interpretation of conditional promises and threats. Canadian Journal of Experimental Psychology 58, 106–112 (2004)

On the Acceptability of Incompatible Arguments

Souhila Kaci[1], Leendert van der Torre[2], and Emil Weydert[2]

[1] CRIL, Rue de l'Université SP 16 62307 Lens Cedex, France
kaci@cril.univ-artois.fr
[2] Computer Science and Communication, University of Luxembourg, Luxembourg
{leon.vandertorre, emil.weydert}@uni.lu

Abstract. In this paper we study the acceptability of incompatible arguments within Dung's abstract argumentation framework. As an example we introduce an instance of Dung's framework where arguments are represented by propositional formulas and an argument attacks another one when the conjunction of their representations is inconsistent, which we characterize as a kind of symmetric attack. Since symmetric attack is known to have the drawback to collapse the various argumentation semantics, we consider also two variations. First, we consider propositional arguments distinguishing support and conclusion. Second, we introduce a preference ordering over the arguments and we define the attack relation in terms of a symmetric incompatibility relation and the preference relation. We show how to characterize preference-based argumentation using a kind of acyclic attack relation.

1 Introduction

Dung's abstract argumentation framework [10] is based only on sets (whose elements are called *arguments*) carrying a binary relation (called the *attack* relation). Due to this abstract perspective, it can and has been used in several ways, for example as a general framework for non-monotonic reasoning, for argumentation, and as a component in agent communication, dialogue, or decision making.

However, the increasing popularity of argumentation-based systems has revealed a dilemma. On the one hand many users appreciate the abstract arguments of Dung's framework, enabling them to reason about arguments without being forced to use a prescribed representation such as rules. On the other hand, in general they find it difficult to think in terms of attack relations. For example, does the argument

> "The soccer game is going to be very interesting, many people will watch the game because it is Barcelona against Arsenal"

attack the argument

> "The soccer game is going to be boring, because the first half was already won by 6-0"

or vice versa? Typically, users specify only whether two arguments are incompatible, i.e., whether two arguments cannot be held together. Incompatibility means here simply

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 247–258, 2007.

that a person cannot forward the two arguments without contradicting himself. This means that the attack relation is symmetric, because if argument A is incompatible with argument B, then argument B is incompatible with argument A.

In this paper we are therefore interested in incompatibility of arguments. However, Besnard and Hunter [7] and Coste-Marquis *et al.* [9] have shown that symmetric attack relations – as represented by propositional argumentation – are not very expressive in the sense that the various semantics of Dung's argumentation framework collapse to selecting either a maximal conflict-free subset of arguments, or the intersection of the maximal conflict-free subsets. We therefore propose that argumentation based systems should use the following argumentation specification:

> **Argumentation Specification for Argumentation Based Systems.** The user specifies abstract arguments, a symmetric incompatibility relation on the arguments, and a preference relation over the arguments. The system calculates first the attack relation from the incompatibility relation and the preference relation, and thereafter the acceptable arguments using one of Dung's semantics.

This proposed specification leaves us the freedom to define the properties of the preference relation and the way to define the attack relation from the other two relations. For the preference relation most users prefer to use a transitive relation, so, for example, a partial order or a total order. For the combination we can define, for example, that argument A attacks argument B when A and B are incompatible and A is at least as preferred as B [11], or B is not preferred to A [2,5].

The choice among these alternatives may depend on which alternative is most intuitive for the user, but it seems that there is no strong preference for either of them. It may depend also on the expressive power of the alternatives: if one representation can represent a larger range of attack relations, it may be a better basis for argumentation based systems. For example, do we again have a collapse of the various semantics, as in the case of symmetric attack relations without a preference relation, or can we represent any kind of attack relation and do we thus cover the full range of possibilities? As we show later in this paper, the answer is often somewhere in the middle: we do not have a collapse of the various semantics, but we can represent only a subset of all possible attack relations.

In this paper we are therefore interested in characterizing various kinds of alternative argumentation frameworks, in the following sense. Given an argumentation framework consisting of a set of arguments and one or more other mathematical elements, we define a mapping from the argumentation framework to a Dung's abstract argumentation framework. Then we say that the alternative framework is characterized by a property if the following two conditions hold. First, for every possible mapping, the property holds for Dung's framework. Second, for every instance of Dung's framework satisfying the property, there is an instance of the alternative framework that is mapped onto it. We say also that Dung's theory together with the property is represented by the alternative framework.

For example, we consider what we call a propositional argumentation framework in which each argument A is represented by a formula $prop(A)$ from propositional logic, and we define the mapping to Dung's framework as follows: argument A attacks argument B if $prop(A) \wedge prop(B)$ is inconsistent. We then characterize the propositional

argumentation theory by symmetric attack, if argument A attacks argument B, then argument B attacks argument A, together with the property that if argument A attacks itself, then it attacks all other arguments too. Thus, this particular kind of symmetric attack can be represented by propositional argumentation.

In particular, in this paper we address the following two questions:

1. Can the collapse of the argumentation semantics be avoided when we represent arguments by pairs $\langle H, h\rangle$, where H is a set of propositional formulas supporting the formula h (i.e., H logically implies h in propositional logic), a kind of propositional argumentation promoted, for example, by Amgoud and Cayrol [2]?
2. How do we characterize an argumentation framework with an incompatibility relation together with a preference relation over arguments, and the mapping that argument A attacks argument B if and only if A and B are incompatible, and B is not preferred to A, as suggested by Amgoud and Cayrol [2] and Bench-Capon [5]?

The results in this paper show that the attack relations which cannot be represented contain a particular kind of cycles of attacking arguments. Attack cycles among arguments have raised already considerable attention in the argumentation literature, in particular due to the fact that odd cycles have a distinct behavior from even cycles, in the sense that the former arguments are not contained in any acceptable preferred or stable semantics, whereas some of the latter are. However, the problem with this distinct behavior is that for users it is very hard to distinguish, for example, a 5-cycle from a 6-cycle. The fact that such cycles can no longer be represented, may be an alternative explanation why it is easier for users to represent an incompatibility relation together with a preference relation, than an attack relation.

The layout of this paper is as follows. In Section 2 we introduce Dung's framework, propositional and extended propositional argumentation, and we characterize them. In Section 3 we introduce preference-based argumentation and we characterize it.

2 Propositional Argumentation

In this section we repeat Dung's argumentation framework, we discuss basic propositional argumentation and we characterize it as a kind of symmetric attack. Then we discuss an extended form of propositional argumentation and we characterize it. Finally we introduce and discuss a notion of closure under supported arguments, and we show that under this assumption extended propositional argumentation collapses to propositional argumentation.

2.1 Dung's Framework

Argumentation is a reasoning model which consists of constructing arguments, determining potential conflicts between arguments, and selecting acceptable arguments. Dung's framework [10] is based on a binary attack relation among arguments.

Definition 1 (Argumentation Framework). *An argumentation framework is a tuple $\langle \mathcal{A}, \mathcal{R}\rangle$ where $\mathcal{A}$ is a set of arguments and $\mathcal{R}$ is a binary attack relation on $\mathcal{A} \times \mathcal{A}$.*

The semantics of Dung's argumentation framework is based on the two notions of defence and conflict-freeness.

Definition 2 (Defence). *A set of arguments $\mathcal{S}$ defends an argument A iff for each argument B of $\mathcal{A}$ which attacks A, there is an argument C in $\mathcal{S}$ which attacks B.*

Definition 3 (Conflict-Free). *A set of arguments $\mathcal{S}$ is conflict-free iff there are no $A, B \in \mathcal{S}$ such that $A\mathcal{R}B$.*

The following definition summarizes various semantics of acceptable arguments proposed in the literature. The output of the argumentation framework is derived from the set of acceptable arguments which are selected with respect to an acceptability semantics.

Definition 4 (Acceptability Semantics). *Let $\mathcal{S} \subseteq \mathcal{A}$.*

- *$\mathcal{S}$ is* admissible *iff it is conflict-free and defends all its elements.*
- *A conflict-free $\mathcal{S}$ is a* complete extension *iff $\mathcal{S} = \{A \mid \mathcal{S}$ defends $A\}$.*
- *$\mathcal{S}$ is a* grounded extension *iff it is the smallest (for set inclusion) complete extension.*
- *$\mathcal{S}$ is a* preferred extension *iff it is a maximal (for set inclusion) complete extension.*
- *$\mathcal{S}$ is a* stable extension *iff it is a preferred extension that attacks all arguments in $\mathcal{A}\backslash\mathcal{S}$.*

Many properties and relations among these semantics have been studied by Dung and others.

2.2 Basic Propositional Argumentation

In basic propositional argumentation, each argument is represented by a propositional formula.

Definition 5. *Let L be a language of propositional logic. A basic propositional argumentation framework is a tuple $\langle \mathcal{A}, prop \rangle$ where $\mathcal{A}$ is a set of arguments and $prop$ is a function from $\mathcal{A}$ to L.*

An argument represented by a propositional formula p attacks an argument represented by q if and only if $p \wedge q$ is inconsistent.

Definition 6. *$\langle \mathcal{A}, prop \rangle$ represents $\langle \mathcal{A}, \mathcal{R} \rangle$ if and only if for all $A, B \in \mathcal{A}$, we have $A \mathcal{R} B$ iff $prop(A) \wedge prop(B)$ is inconsistent in propositional logic. We say also that $\mathcal{R}$ is represented by $prop$.*

Example 1. Consider a propositional argumentation framework $\langle \{A, B, C, D\}, prop \rangle$ with $prop(A) = p, prop(B) = \neg p \wedge q, prop(C) = q$ and $prop(D) = r \wedge \neg r$. We have that A attacks B and vice versa, D attacks all other arguments and vice versa, and no other attack relations hold.

Roughly, basic propositional argumentation corresponds to symmetric argumentation.

Theorem 1. *If $\mathcal{A}$ is finite, then $\langle \mathcal{A}, \mathcal{R} \rangle$ can be represented by a basic propositional argumentation framework if and only if the following two properties hold:*

1. $\mathcal{R}$ *is symmetric.*
2. *For all arguments* A, B*, if* $A\mathcal{R}A$*, then* $A\mathcal{R}B$*.*

Proof. Soundness. Let $\langle \mathcal{A}, prop \rangle$ *be a basic propositional argumentation theory representing* $\langle \mathcal{A}, \mathcal{R} \rangle$*. Symmetry holds, because* $prop(A) \wedge prop(B)$ *is inconsistent if and only if* $prop(B) \wedge prop(A)$ *is inconsistent (with respect to propositional logic). Moreover, an argument* A *attacks itself,* $A\mathcal{R}A$*, if and only if* $prop(A)$ *is inconsistent. However, if* $prop(A)$ *is inconsistent, then* $prop(A) \wedge prop(B)$ *is inconsistent as well, thus* $A\mathcal{R}B$*.*

Completeness. We prove it by construction. Let $\langle \mathcal{A}, \mathcal{R} \rangle$ *be an arbitrary argumentation theory satisfying that* $\mathcal{R}$ *is symmetric and* $\forall A, B \in \mathcal{A}$*, if* $A\mathcal{R}A$ *then* $A\mathcal{R}B$*. Let* $\langle A_1, \ldots, A_{|\mathcal{A}|} \rangle$ *be a sequence of all the elements of* $\mathcal{A}$*. Moreover, consider* $|\mathcal{A}|$ *propositional atoms* p_i*, define*

$$P = \bigwedge_{1 \leq i,j \leq |\mathcal{A}|} \{\neg(p_i \wedge p_j) \mid A_i \mathcal{R} A_j\},$$

$$prop(A_i) = p_i \wedge P.$$

We prove that this basic argumentation framework represents $\langle \mathcal{A}, \mathcal{R} \rangle$ *by showing that* $\forall A, B \in \mathcal{A}$*,* $A\mathcal{R}B$ *if and only if* $prop(A) \wedge prop(B)$ *is inconsistent.*

Let p *and* q *be the propositional atoms associated to* A *and* B *respectively. Suppose that* $A\mathcal{R}B$*. Then* $prop(A) = p \wedge P$ *and* $prop(B) = q \wedge P$*. Since* $A\mathcal{R}B$ *then* $\neg(p \wedge q)$ *belongs to* P*. Thus* $prop(A) \wedge prop(B)$ *is inconsistent.*

Suppose that $prop(A) \wedge prop(B)$ *is inconsistent. This means that* $(p \wedge P) \wedge (q \wedge P)$ *is inconsistent. Since* p *and* q *are propositional atoms and thus positive formulas, this can hold only if either:*

1. $\neg(p \wedge q)$ *belongs to* P*, which means that* $A\mathcal{R}B$*. By symmetry of* $prop(A) \wedge prop(B)$ *we have also* $B\mathcal{R}A$*.*
2. *We have* $p_i = p_j = p$ *or* $p_i = p_j = q$ *for some* p_i *and* p_j*, which means either* A *or* B *attacks itself. But then by the second rule we have either* $A\mathcal{R}B$ *or* $B\mathcal{R}A$*, and by symmetry we have in both cases* $A\mathcal{R}B$ *and* $B\mathcal{R}A$*.*

Consequently, in both cases we have $A\mathcal{R}B$*. This concludes the proof.*

2.3 Symmetric Attack Relations

Besnard and Hunter [7] and Coste-Marquis *et al.* [9] show that only two distinct forms of acceptability are possible when the considered frameworks are symmetric. Those forms of acceptability are quite rudimentary, but tractable; this contrasts with the general case where all the forms of acceptability are intractable (except the ones based on grounded extensions).

Theorem 2. *[9] Let* $\langle \mathcal{A}, \mathcal{R} \rangle$ *be an argumentation framework where* $\mathcal{R}$ *is symmetric, non-empty, and irreflexive.* $A \in \mathcal{A}$ *belongs to every preferred (equivalently, stable) extension of* $\langle \mathcal{A}, \mathcal{R} \rangle$ *if and only if there is no* $B \in \mathcal{A}$ *such that* $B\mathcal{R}A$*.*

Coste-Marquis *et al.* summarize that there are at most two distinct forms of acceptability for symmetric argumentation frameworks: all the forms of skeptical acceptability coincide with the notion of acceptability with respect to the grounded extension, and credulous acceptability with respect to preferred extensions coincides with credulous acceptability with respect to stable extensions.

Consequently, one has to consider more general acceptability notions if one wants to get more than one semantics, as we do here; indeed, skeptical acceptability is rather poor since it characterizes as acceptable only those arguments which are not attacked. For these reasons, Coste-Marquis *et al.* turn to acceptability concepts for *sets* of arguments, that is, the question is to determine whether or not it is reasonable to accept some arguments together.

2.4 Extended Propositional Argumentation

Arguments can be represented by pairs $\langle H, h\rangle$, where $H \cup \{h\}$ is a set of propositional formulas [12]. We call H the support and h the conclusion of the argument.

Definition 7. *Let L be a language of propositional logic. An extended propositional argumentation framework is a tuple $\langle \mathcal{A}, sup, con\rangle$ where $\mathcal{A}$ is a set (of arguments), $sup : \mathcal{A} \to 2^L$ is a (support) function from arguments to sets of formulas, and $con : \mathcal{A} \to L$ is a (conclusion) function from arguments to formulas such that for all $A \in \mathcal{A}$, $sup(A)$ (proposition) logically implies $con(A)$.*

We have two kinds of attack. An argument $\langle H, h\rangle$ rebuts $\langle K, k\rangle$ if and only if $h \wedge k$ is inconsistent, and the former undercuts the latter if and only if $\{h\} \cup K$ is inconsistent. Note that using these definitions of rebut and undercut, due to the fact that $sup(A)$ logically implies $con(A)$, we have that if A rebuts B, then also A undercuts B (but not necessarily vice versa). We therefore consider undercutting only in the following definition of representation.

Definition 8. *$\langle \mathcal{A}, sup, con\rangle$ represents $\langle \mathcal{A}, \mathcal{R}\rangle$ if and only if for all $A, B \in \mathcal{A}$, we have $A\,\mathcal{R}\,B$ iff $\{con(A)\} \cup sup(B)$ is inconsistent in propositional logic.*

Example 2. Let $\langle \{A, B, C\}, sup, con\rangle$ be an extended propositional argumentation theory with $sup(A) = \{\neg p \wedge q\}$, $con(A) = q$, $sup(B) = \{p \wedge \neg q\}$ and $con(B) = p$, $sup(C) = \{p \vee r, \neg r \vee s, \neg s\}$ and $con(C) = p$. We may represent this theory also by $A = \langle \{\neg p \wedge q\}, q\rangle$, $B = \langle \{p \wedge \neg q\}, p\rangle$, and $C = \langle \{p \vee r, \neg r \vee s, \neg s\}, p\rangle$. Note that $\langle \{\neg p \wedge q\}, q\rangle$ is an argument whereas $\langle \{\neg p, q\}, q\rangle$ is not due to the minimality criterion of the support of an argument. We have argument A attacks argument B and vice versa, argument C attacks argument A, and no other attack relations hold.

Theorem 3. *If $\mathcal{A}$ is finite, then $\langle \mathcal{A}, \mathcal{R}\rangle$ can be represented by an extended propositional argumentation framework if and only if for all arguments $A, B \in \mathcal{A}$, if $A\mathcal{R}A$, then $B\mathcal{R}A$.*

Proof. Soundness. Let $\langle \mathcal{A}, sup, con\rangle$ be an extended propositional argumentation framework representing $\langle \mathcal{A}, \mathcal{R}\rangle$. Suppose that $A\mathcal{R}A$. Then $\{con(A)\} \cup sup(A)$ is inconsistent. Following Definition 7, $sup(A)$ logically implies $con(A)$. Hence, $sup(A)$

must be inconsistent ($con(A)$ is not necessarily inconsistent). But then we have for all $B \in \mathcal{A}$ that $\{con(B)\} \cup sup(A)$ is inconsistent, i.e., that $B\mathcal{R}A$.

Completeness. We prove it by construction. Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an arbitrary argumentation framework satisfying the condition that if $A\mathcal{R}A$, then $B\mathcal{R}A$. Let $\langle A_1, \ldots, A_{|\mathcal{A}|} \rangle$ be a sequence of all the elements of $\mathcal{A}$. Moreover, consider $|\mathcal{A}|$ propositional positive atoms p_i, define

$$P_i = \{\neg(p_i \wedge p_j) \mid 1 \leq j \leq |\mathcal{A}|, A_j \mathcal{R} A_i\},$$

$$sup(A_i) = \{p_i\} \cup P_i,$$

$$con(A_i) = p_i.$$

We prove that the above extended propositional framework represents the argumentation framework $\langle \mathcal{A}, \mathcal{R} \rangle$, i.e. $\forall A, B \in \mathcal{A}$, $A\mathcal{R}B$ iff $\{con(A)\} \cup sup(B)$ is inconsistent.

Suppose that $A\mathcal{R}B$. Let p and q be the propositional atoms associated to A and B respectively. We have conclusion $con(A) = p$ and support $sup(B) = \{q\} \cup P$ with $P = \{\neg(q \wedge p_j) \mid A_j \mathcal{R} B\}$. We have that $\neg(q \wedge p) \in P$ due to $A\mathcal{R}B$. Then it follows that $\{con(A)\} \cup sup(B)$ is inconsistent.

Suppose now that $\{con(A)\} \cup sup(B)$ is inconsistent. This means that we have that $\{p, q\} \cup \{\neg(q \wedge p_j) \mid A_j \mathcal{R} B\} = \{p, q\, \neg(q \wedge p_1), \neg(q \wedge p_2), \cdots, \neg(q \wedge p_m)\}$ is inconsistent. Since p, q and p_j are propositional atoms, and thus positive formulas, this formula is inconsistent if and only if either:

1. *there exists p_j ($j = 1, \cdots, m$) such that $p_j = p$. Then $A\mathcal{R}B$, or*
2. *there exists p_j ($j = 1, \cdots, m$) such that $p_j = q$. Then $B\mathcal{R}B$, and therefore due to the second property $A\mathcal{R}B$.*

Hence, in both cases we have $A\mathcal{R}B$. This concludes the proof.

2.5 Closure Under Supported Formulas

In propositional argumentation, we may have, for example, $\langle \{p \wedge q\}, p \rangle$ as an argument, without having $\langle \{p \wedge q\}, q \rangle$ as an argument. However, in some cases the set of arguments of an extended propositional theory has been generated in some way. In those cases, we may want to consider some closure conditions.

Definition 9. *Let $\langle \mathcal{A}, sup, con \rangle$ be an extended propositional argumentation framework.*

- *A set of arguments $S \subseteq \mathcal{A}$ is said to be* closed under supported formulas *iff for all arguments $A \in \mathcal{S}$ and all propositional formulas p logically implied by $sup(A)$ but not by a strict subset of $sup(A)$, there is an argument $B \in \mathcal{S}$ such that $sup(B) = sup(A)$ and $con(B) = p$.*
- *We say that $\langle \mathcal{A}, sup, con \rangle$ is closed under supported formulas if $\mathcal{A}$ is.*

If an argumentation framework is closed under supported formulas, then its complete extensions are closed as well.

Lemma 1. *If the argument $\langle H, h \rangle$ is in a complete extension, then $\langle H, \wedge H \rangle$ is also in the extension.*

Lemma 2. *If the argument $\langle H, \wedge H\rangle$ is in a complete extension, then $\langle H, h\rangle$ is in the extension for all h logically implied by H, but not by a strict subset of H.*

Theorem 4. *If an extended propositional argumentation framework is closed under supported formulas, then its complete extensions are closed under them as well.*

A consequence of Theorem 4 is that under closure of supported formulas, extended propositional argumentation framework can be reduced to propositional argumentation framework. Consequently, under this assumption, we have again a collapse of the argumentation semantics.

Example 3. Suppose we have an argument $\langle\{p \wedge q\}, p\rangle$ in a complete extension. Then the lemmas show that we have also $\langle\{p \wedge q\}, p \wedge q\rangle$ (lemma 1) and thus $\langle\{p \wedge q\}, q\rangle$ (lemma 2). But then, clearly, there is no use of arguments like $\langle\{p \wedge q\}, p\rangle$. We can restrict ourselves to arguments $\langle\{p \wedge q\}, p \wedge q\rangle$ (under the assumption of closure of suported formulas, of course).

Thus, under the assumption of closure under supported formulas, the distinction between support and conclusion is just syntactic sugar.

3 Preferences Among Arguments

In this section we consider the extension of symmetric argumentation with preferences. We start with some definitions concerning preferences.

Definition 10. *A (partial) pre-order on a set $\mathcal{A}$, denoted $\succeq$, is a reflexive and transitive relation. $\succeq$ is said to be total if for all $A, B \in \mathcal{A}$ we have $A \succeq B$ or $B \succeq A$. $\succ$ denotes the strict order associated with $\succeq$, i.e., $A \succ B$ iff $A \succeq B$ and not $B \succeq A$.*

A preference relation on $\mathcal{A}$ is a pre-order $\succeq$ on $\mathcal{A}$ such that $\forall A, B \in \mathcal{A}$, $A \succeq B$ (resp. $A \succ B$) expresses that A is at least as preferred as (resp. strictly preferred to) B.

The new preference-based argumentation framework uses an incompatibility and a preference relation. The incompatibility relation should not be interpreted as an attack relation, since incompatibility relations are always symmetric, while attack relations are often asymmetric.

Definition 11 (Incompatibility+Preference Argumentation Framework). *An incompatibility+preference argumentation framework is a triplet $\langle \mathcal{A}, \mathcal{C}, \succeq\rangle$ where $\mathcal{A}$ is a set of arguments, $\mathcal{C}$ is a symmetric binary incompatibility relation on $\mathcal{A} \times \mathcal{A}$, and $\succeq$ is a preference relation on $\mathcal{A} \times \mathcal{A}$.*

Starting with a set of arguments, a symmetric incompatibility relation, and a preference relation, we exploit the latter two for specifying a Dung-style attack relation. Then we use an arbitrary semantics of Dung to characterize the set of acceptable arguments. In contrast to most other approaches [2,13] (but see [4,5] for exceptions), our approach to reasoning about preferences in argumentation does not refer to the internal structure of the arguments. The use of a symmetric incompatibility relation makes sense in many applications such as dialogue when the internal structure of the arguments is not known

and thus does not allow to know whether the attack relation is undercut or rebut. Instead we may only know that the two arguments cannot be used together, i.e., they are incompatible.

Definition 12. *Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework and $\langle \mathcal{A}, \mathcal{C}, \succeq \rangle$ an incompatibility+preference argumentation framework. We say that $\langle \mathcal{A}, \mathcal{C}, \succeq \rangle$ represents $\langle \mathcal{A}, \mathcal{R} \rangle$ iff for all arguments A and B of $\mathcal{A}$, we have $A \mathcal{R} B$ iff $A \mathcal{C} B$ and not $B \succ A$. We say also that $\mathcal{R}$ is represented by $\mathcal{C}$ and $\succeq$.*

Example 4. Let $\langle \{A, B\}, \{A \mathcal{C} B, B \mathcal{C} A\}, \{A \succeq B\} \rangle$. A attacks B but not vice versa.

Since the attack relation is defined from the incompatibility and the preference relation, the other notions introduced by Dung can be applied also to the incompatibility+preference argumentation framework. For example, to determine the grounded semantics of the incompatibility+preference framework, we first compute the attack relation, and then the grounded semantics as in the general case.

An acyclic/symmetric argumentation framework is an argumentation framework in which the attack relation is acyclic/symmetric, *etc*. In this paper we define an acyclic strict attack relation as follows. If there is a strict attack path where argument A_1 attacks argument A_2 but not vice versa, argument A_2 attacks argument A_3 but not vice versa, ..., then argument A_n does not attack argument A_1.

Definition 13 (Acyclic Argumentation Framework). *An argument A strictly attacks B if A attacks B and B does not attack A. A strict acyclic argumentation framework is an argumentation framework $\langle \mathcal{A}, \mathcal{R} \rangle$ in which there is no sequence of arguments $\langle A_1, \ldots, A_n \rangle$ such that A_1 strictly attacks A_2, A_2 strictly attacks A_3, ..., A_{n-1} strictly attacks A_n, and A_n attacks A_1.*

To prove that acyclic attacks can be characterized by incompatibilities and preferences, we have to show that the implication holds in both ways. We start with the implication from right to left.

Lemma 3. *If the incompatibility+preference argumentation framework $\langle \mathcal{A}, \mathcal{C}, \succeq \rangle$ represents the argumentation framework $\langle \mathcal{A}, \mathcal{R} \rangle$, then $\langle \mathcal{A}, \mathcal{R} \rangle$ is a strictly acyclic argumentation framework (in the sense of Definition 13).*

Proof. We prove the lemma by contradiction. Assume there exists an incompatibility+preference argumentation framework $\langle \mathcal{A}, \mathcal{C}, \succeq \rangle$ representing argumentation framework $\langle \mathcal{A}, \mathcal{R} \rangle$ such that $\langle \mathcal{A}, \mathcal{R} \rangle$ is not a strictly acyclic argumentation framework. In other words, there exists a sequence of strictly attacking arguments $\langle A_1, \ldots, A_n \rangle$ with $A_n \mathcal{R} A_1$. Since $\langle \mathcal{A}, \mathcal{C}, \succeq \rangle$ represents $\langle \mathcal{A}, \mathcal{R} \rangle$, we have $A_i \mathcal{C} A_{i+1}$ and not $A_{i+1} \succ A_i$. Due to symmetry of $\mathcal{C}$, we have also $A_{i+1} \mathcal{C} A_i$. Since the attacks are strict, we do not have not $A_i \succ A_{i+1}$, and we thus have $A_i \succ A_{i+1}$. Moreover, due to transitivity of $\succeq$, we have $A_1 \succ A_n$. This implies that we cannot have $A_n \mathcal{R} A_1$. Contradiction, thus the lemma holds.

Now we show that the implication from left to right holds. We prove this lemma by construction: given an acyclic argumentation framework, we construct an incompatibility+preference framework representing it.

Lemma 4. *If $\langle \mathcal{A}, \mathcal{R} \rangle$ is a strictly acyclic argumentation framework, then there is an incompatibility+preference argumentation framework $\langle \mathcal{A}, \mathcal{C}, \succeq \rangle$ that represents it.*

Proof. By construction. Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be a strictly acyclic argumentation framework. Moreover, consider an incompatibility+preference argumentation framework $\langle \mathcal{A}, \mathcal{C}, \succeq \rangle$ defined as follows:

- *$\mathcal{C} = \{(A, B) \mid A\mathcal{R}B$ or $B\mathcal{R}A\}$ is the symmetric closure of $\mathcal{R}$*
- *$\succeq$ is the transitive and reflexive closure of the strict attack relations in $\mathcal{R}$, i.e., the transitive and reflexive closure of $\{(A, B) \mid A\mathcal{R}B$ and not $B\mathcal{R}A\}$*

We show that $\forall A, B \in \mathcal{A}$, $A\mathcal{R}B$ if and only if $A\mathcal{C}B$ and not $B \succ A$. From left to right, we show that $\forall A, B \in \mathcal{A}$ if $A\mathcal{R}B$ then $A\mathcal{C}B$ and not $B \succ A$. Suppose that $A\mathcal{R}B$ and not $A\mathcal{C}B$ or $B \succ A$. By construction we have $A\mathcal{C}B$ due to $A\mathcal{R}B$, and therefore we have $B \succ A$. By construction $B \succ A$ means that there is a sequence of strict attacks $B\mathcal{R}A_1$, $A_1\mathcal{R}A_2$, …, $A_n\mathcal{R}A$. Consequently, due to the acyclicity property, we do not have $A\mathcal{R}B$. Contradiction.

From right to left, we show that $\forall A, B \in \mathcal{A}$ if $A\mathcal{C}B$ and not $B \succ A$ then $A\mathcal{R}B$. Suppose now that $A\mathcal{C}B$, not $B \succ A$ and not $A\mathcal{R}B$. By construction $A\mathcal{C}B$ means that either $A\mathcal{R}B$ or $B\mathcal{R}A$ holds. Since $A\mathcal{R}B$ does not hold by hypothesis we have $B\mathcal{R}A$. Also by construction $B\mathcal{R}A$ and not $A\mathcal{R}B$ implies that $B \succeq A$. $B \succeq A$ and not $B \succ A$ implies $A \succeq B$. Since $A \succeq B$ and not $A\mathcal{R}B$, $A \succeq B$ must be added by reflexive or transitive closure, and therefore there must be a sequence of strict attacks $A\mathcal{R}A_1$, $A_1\mathcal{R}A_2$, …, $A_n\mathcal{R}B$. Consequently, due to the acyclicity property, we do not have $B\mathcal{R}A$. Contradiction.

Summarizing, strictly acyclic argumentation frameworks are characterized by incompatibility+preference argumentation frameworks.

Theorem 5. *$\langle \mathcal{A}, \mathcal{R} \rangle$ is a strictly acyclic argumentation framework (in the sense of Definition 13) if and only if there is an incompatibility+preference argumentation framework $\langle \mathcal{A}, \mathcal{C}, \succeq \rangle$ that represents it (in the sense of Definition 12).*

4 Related Work

Further developments of Dung's framework have been studied along various directions:

- Dung's abstract framework has been used mainly in combination with more detailed notions of arguments, for example arguments consisting of rules, or arguments consisting of a justification and a conclusion.
- Analogously, various kinds of attack relations have been distinguished, such as rebutting and undercutting.
- Constraints have been imposed on the attack relations, such as symmetry in symmetric argumentation frameworks [9].
- There have been several attempts to modify or generalize Dung's framework, for example by introducing preferences [2], priorities [13], values [5], or collective arguments [8].

We believe that our approach of characterizing argumentation frameworks with properties of Dung's attack relation is a powerful way to relate these various approaches in argumentation theory. For example, though various authors present their argumentation framework as an extension of Dung's framework, which has the technical consequence that they define also new notions of, for example, defence and acceptance, we can consider also their argumentation framework as an alternative representation of Dung's framework, which has the consequence that we do not have to introduce such notions.

The main results of this paper are concerned with preference-based argumentation. Simari and Loui [14] introduce preference relations over arguments, and various proposals have been made how to specify and compute these preferences. The authors of [13,15] consider arguments composed of defeasible rules, and they use the argument *structure* to derive preference relations. For instance, one argument is more specific about the current evidence than the other one, which makes the first argument stronger. Alternatively, several authors [6,3,1] have built arguments from beliefs tagged with explicit priorities, such as certainty levels. The arguments using higher-level beliefs are considered stronger than those using lower-level beliefs. Bench-Capon [5] does not consider the structure of arguments but derives a preference ordering from the values they promote. Since arguments promote only a single value, an argument is better than another one if and only if the value promoted by the former is preferred to the value promoted by the latter argument.

In [11] we consider a representation of Dung's framework by an incompatibility and preference argumentation framework, where A attacks B if and only if A and B are incompatible, and A is at least as preferred as B. The acyclicity or loop condition that characterizes this kind of argumentation framework is that if there is a cycle in the sense that A_1 attacks A_2, ..., A_{n-1} attacks A_n, A_n attacks A_1, then we have that A_2 attacks A_1, ..., A_n attacks A_{n-1}, A_1 attacks A_n. The representation used in this paper is much more widely used, and the loop condition of this paper looks more natural than the condition of [11].

5 Summary

Dung's abstract argumentation framework [10] is based only on sets of arguments carrying a binary attack relation. In this paper we develop an abstract framework for incompatible arguments within Dung's abstract argumentation framework. As an example we introduce an instance of Dung's framework where arguments are represented by propositional formulas and an argument attacks another one when the conjunction of their representations is inconsistent, which we characterize as a kind of symmetric attack. Since symmetric attack is known to have the drawback to collapse the various argumentation semantics, we consider two variations.

First, we consider propositional arguments distinguishing support and conclusion, and consider the question whether the collapse of the argumentation semantics be avoided when we represent arguments by pairs $\langle H, h\rangle$, where H is a set of propositional formulas supporting the formula h (i.e., H logically implies h in propositional logic), a kind of propositional argumentation introduced by Amgoud and Cayrol [2].

We show it is nearly as expressive as Dung's framework. However, we show also that when we extend it with a property which we call closure under supported formulas, then a similar collapse arises.

Second, we consider an argumentation framework with an incompatibility relation together with a preference relation over arguments, and the mapping that argument A attacks argument B if and only if A and B are incompatible, and B is not preferred to A, as suggested by Amgoud and Cayrol [2] and Bench-Capon [5]. We characterize the attack relation by a particular kind of loop condition. If there is a sequence of strict attacks $\langle A_1, \ldots, A_n \rangle$, i.e., A_1 attacks A_2 but not vice versa, A_2 attacks A_3 but not vice versa, etc, then A_n does not attack A_1.

References

1. Amgoud, L., Cayrol, C.: Inferring from inconsistency in preference-based argumentation frameworks. International Journal of Approximate Reasoning 29(1), 125–169 (2002)
2. Amgoud, L., Cayrol, C.: A reasoning model based on the production of acceptable arguments. Annals of Mathematics and Artificial Intelligence 34, 197–216 (2002)
3. Amgoud, L., Cayrol, C., LeBerre, D.: Comparing arguments using preference orderings for argument-based reasoning. In: 8th International Conference on Tools with Artificial Intelligence (ICTAI'96), pp. 400–403 (1996)
4. Amgoud, L., Parsons, S., Perrussel, L.: An argumentation framework based on contextual preferences. In: Proceedings of the 3rd International Conference on Formal and Applied Practical Reasoning (FAPR'00), pp. 59–67 (2000)
5. Bench-Capon, T.: Persuasion in practical argument using value based argumentation framework. Journal of Logic and Computation 13(3), 429–448 (2003)
6. Benferhat, S., Dubois, D., Prade, H.: Argumentative inference in uncertain and inconsistent knowledge base. In: 9th Int. Conf. on Uncertainty in AI (UAI'93), pp. 411–419 (1993)
7. Besnard, P., Hunter, A.: A logic-based theory of deductive arguments. Artificial Intelligence 128, 203–235 (2001)
8. Bochman, A.: Propositional Argumentation and Causal Reasoning. In: 11th International Joint Conference on Artificial Intelligence (IJCAI'05), pp. 388–393 (2005)
9. Coste-Marquis, S., Devred, C., Marquis, P.: Symmetric argumentation frameworks. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 317–328. Springer, Heidelberg (2005)
10. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77, 321–357 (1995)
11. Kaci, S., van der Torre, L., Weydert, E.: Acyclic Argumentation: Attack = Conflict + Preference. In: Proceedings of the 17th European Conference on Artificial Intelligence (ECAI 2006), pp. 725–726 (2006)
12. Pollock, J.L.: Defeasible reasoning. Cognitive Science 11(4), 481–518 (1987)
13. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorties. Journal of Applied Non-Classical Logics 7, 25–75 (1997)
14. Simari, G.R., Loui, R.P.: A mathematical treatment of defeasible reasoning and its implementation. Artificial Intelligence 53, 125–157 (1992)
15. Stolzenburg, F., García, A.J., Chesñevar, C.I., Simari, G.R.: Computing generalized specificity. Journal of Applied Non-Classical Logics 13(1), 87–113 (2003)

Handling Ignorance in Argumentation: Semantics of Partial Argumentation Frameworks

C. Cayrol[1], C. Devred[2,3], and M.C. Lagasquie-Schiex[1]

[1] IRIT, Université Paul Sabatier, Toulouse
{ccayrol, lagasq} @irit.fr
[2] CRIL, Université d'Artois, Lens
devred@cril.univ-artois.fr
[3] LERIA, Université d'Angers
caroline.devred@info.univ-angers.fr

Abstract. In this paper we propose semantics for acceptablity in partial argumentation frameworks (PAF). The PAF is an extension of Dung's argumentation framework and has been introduced in [1] for merging argumentation frameworks. It consists in adding a new interaction between arguments representing the ignorance about the existence of an attack.

The proposed semantics are built following Dung's method, so that they generalize Dung's semantics without increasing the temporal complexity.

1 Introduction

Argumentation has become an influential approach to treat AI problems including defeasible reasoning and some forms of dialogue between agents (see e.g. [2,3,4,5,6]). Argumentation is basically concerned with the exchange of interacting arguments. Usually, the interaction takes the form of a conflict, called attack. For example, a logical argument can be a pair ⟨set of assumptions, conclusion⟩, where the set of assumptions entails the conclusion according to some logical inference schema. Then a conflict occurs for instance if the conclusion of an argument contradicts an assumption of another argument. The main issue for any theory of argumentation is the selection of acceptable sets of arguments, based on the way arguments interact. Intuitively, an acceptable set of arguments must be in some sense coherent and strong enough (e.g. able to defend itself against all attacking arguments). It is convenient to explore the concept of acceptability through argumentation frameworks, and especially Dung's framework ([7]), which abstracts from the nature of the arguments, and represents interaction under the form of a binary relation "attack" on a set of arguments. Such an argumentation framework fits well with situations where the knowledge about the interactions can be assumed to be complete. That is, for each pair (a, b) of considered arguments, it is possible to prove that there is an attack, or not.

There are numerous proposals by numerous researchers for capturing forms of argumentation in logic. Many of these logic-based argumentation systems are promising for simulating argumentation forms which arise in the real-world. Nevertheless, it remains some cases in which it is difficult to formalize logically the structure of the arguments. For instance, professionals such as lawyers, journalists, politicians for instance

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 259–270, 2007.

use arguments for analyzing situations before presenting some information to a given audience. An argument in that case can be a piece of text or of a discourse, used for convincing the audience. In other cases, arguments are used for analyzing situations before making some decisions. Such arguments can be just positions that can be advanced for or against options or other positions. More generally, argumentation can be based on any kind of information coming from heterogeneous sources, objective information such as measured or observed values for instance, or subjective information such as beliefs. So, arguments may have no explicit internal logical structure. Consider an agent for which different kinds of information are available. Some kinds of information enable the agent to build logical arguments (for instance a, b, c), while other kinds of information only permit to advance informal arguments (for instance d, e) for trying to defeat other arguments. So, the agent can be sure that a attacks b and a does not attack c, but the agent also wants to take into account an attack of e by d even if nothing can be said about an attack of d by e. Moreover, the generation of arguments often takes place in a resource-bounded environment, and the agent may not have enough time for computing all the possible attacks. So, there is a need for the representation of partial knowledge about the interactions between arguments.

In Dung's framework, there is no space for ignorance. If a given pair of arguments (a, b) does not appear in the graph of the attack relation, it means that certainly a does not attack b. Another interpretation could be given, according to which there is some uncertainty, and even ignorance about an attack of b by a. This interpretation enables to complete the knowledge later on.

In this paper, we propose to investigate a new kind of abstract argumentation framework in order to handle ignorance in argumentation, and to distinguish between certainty of non-attack and ignorance of attack. This new framework, called Partial Argumentation Framework (or PAF), has recently been introduced in ([1,8]) for the particular purpose of merging argumentation systems. However, no formal study of a partial argumentation framework has been made, especially concerning the key concept of acceptability. Such a study is the topic of this paper.

The paper is organized as follows : Dung's abstract framework is recalled in Section 2. Partial argumentation frameworks are defined in Section 3. Section 4 presents the links between classical frameworks and partial frameworks. It is shown that a classical framework can be given different interpretations in terms of a partial framework. On the other hand, it is shown that a partial framework represents a set of classical frameworks. Acceptability in a partial framework is studied in Section 5. The main properties of the proposed semantics are given in Section 6 (in particular some results about temporal complexity).

2 Argumentation Frameworks (AF)

In [7], Dung has proposed an abstract framework for argumentation in which he focuses only on the definition of the status of arguments. For that purpose, he assumes that a set of arguments is given, as well as the different conflicts among them.

We briefly recall that abstract framework:

Definition 1. *An* argumentation framework (AF) *is a pair* $\langle \mathcal{A}, \mathcal{R} \rangle$ *of a set* $\mathcal{A}$ *of arguments and a binary relation* $\mathcal{R}$ *on* $\mathcal{A}$ *called the* attack relation*.* $a_i \mathcal{R} a_j$ *means that* a_i attacks a_j *(or* a_j *is* attacked *by* a_i*). An argumentation framework may be represented by a directed graph, called the* interaction graph*, whose nodes are arguments and edges represent the attack relation.*

In Dung's framework, the *acceptability of an argument* depends on its membership to some sets, called extensions. These extensions are characterised by particular properties. It is a collective acceptability. Let $\mathsf{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework, let $S \subseteq \mathcal{A}$, the main characteristic properties are:

Definition 2. S *is* conflict-free *for* AF *iff there exist no* a_i, a_j *in* S *such that* $a_i \mathcal{R} a_j$.

An argument a *is acceptable w.r.t.* S *for* AF *iff* $\forall b \in \mathcal{A}$ *such that* $b \mathcal{R} a$, $\exists c \in S$ *such that* $c \mathcal{R} b$. S *is* acceptable *for* AF *iff* $\forall a \in S$, a *is acceptable w.r.t.* S *for* AF.

Then several *semantics for acceptability* have been defined in [7]. For instance:

Definition 3. S *is an* admissible set *for* AF *iff* S *is conflict-free and acceptable for* AF.

S *is a* preferred extension *of* AF *iff* S *is maximal for* $\subseteq$ *among the admissible sets for* AF.

3 Partial Argumentation Frameworks (PAF)

In [1,8], an extension of Dung's argumentation framework has been proposed in order to take into account the possible ignorance about the attack between arguments.

Definition 4. *A* partial[1] argumentation framework (PAF) *is a tuple* $\langle \mathcal{A}, \mathcal{R}, \mathcal{I}, \mathcal{N} \rangle$ *where* $\mathcal{A}$ *is a set of arguments and* $\mathcal{R}$*,* $\mathcal{I}$ *and* $\mathcal{N}$ *are three binary relations on* $\mathcal{A}$ *making a partition of* $\mathcal{A} \times \mathcal{A}$*.* $\mathcal{R}$ *represents the* attack relation*,* $\mathcal{N}$ *represents the* non-attack relation *and* $\mathcal{I}$ *represents the* ignorance relation[2].

Intuitively, $(a, b) \in \mathcal{R}$ means that "a certainly attacks b", $(a, b) \in \mathcal{N}$ means that "a certainly does not attack b", $(a, b) \in \mathcal{I}$ means that "the agent does not know the nature of the interaction between a and b". More generally, $(a, b) \in \mathcal{I}$ may be interpreted as "the agent is not certain of the existence of an attack from a to b, but it is possible that this attacks exists". Note that the set of arguments is not assumed finite in the above definition.

In the following, we have chosen to represent a PAF graphically as in [1,8]: an attack (resp. ignorance) from a to b is represented by a plain (resp. dotted) edge from a to b, and the non-attack relation is not explicitly represented[3].

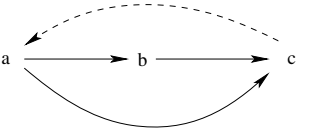
[1] A notion of "partiality" has already been introduced in [9], but with a very different viewpoint (this "partiality" refers to a notion of approximate arguments in order to solve computational issues), clearly not related to our partial argumentation framework.

[2] Though $\mathcal{R}$, $\mathcal{I}$ and $\mathcal{N}$ make a partition of $\mathcal{A} \times \mathcal{A}$, these 3 relations must appear in the tuple because, *a priori*, we do not know which relation can be deduced by the complementation of the two other ones.

[3] Such a representation is judicious in the case there are many non-attacks.

Example 1.

Consider $\mathsf{PAF} = \langle\{a,b,c\}, \{(a,b),(b,c),(a,c)\}, \{(c,a)\}, \{(a,a),(b,b),\ (c,c),(b,a),(c,b)\}\rangle$ represented by:



4 From AF to PAF, and Vice-Versa

4.1 Viewing an AF as a Particular PAF

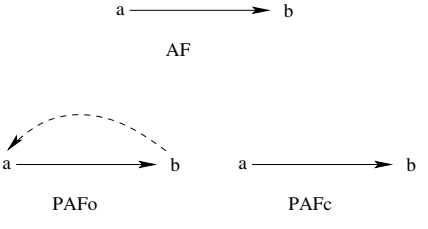
In Dung's framework, $(a,b) \notin \mathcal{R}$ is classically interpreted as "there is no attack from a to b". This interpretation can be said "closed" because there is no doubt about the interaction between a and b. There is an analogy with the well-known Closed World Assumption. So, the closed interpretation considers that an AF is a PAF where $\mathcal{I}$ is the empty set, *i.e.* $\langle\mathcal{A}, \mathcal{R}, \varnothing, \mathcal{A}\times\mathcal{A}\setminus\mathcal{R}\rangle$. This is the more complete interpretation: for each pair (a,b) of arguments, either one knows that a attacks b, or one knows that a does not attack b.

A more cautious interpretation, called "open", considers that there is some ignorance about the interaction from a to b when $(a,b) \notin \mathcal{R}$ and a and b are different. However, intuitively it seems sound to remove the self-attacks[4]. So, $(a,a) \notin \mathcal{R}$ will always be interpreted by $(a,a) \in \mathcal{N}$. So, the open interpretation considers that an AF is a PAF where $\mathcal{N}$ is reduced to the set of pairs (a,a) which are not in $\mathcal{R}$, *i.e.* $\langle\mathcal{A}, \mathcal{R}, \{(a,b)|a \neq b \text{ and } (a,b) \notin \mathcal{R}\}, \{(a,a)|(a,a) \notin \mathcal{R}\}\rangle$.

So, a given AF can be interpreted by two PAFs according to the agent's attitude: PAF_c (closed attitude) or PAF_o (open attitude).

Example 2.

Consider $\mathsf{AF} = \langle\{a,b\}, \{(a,b)\}\rangle$ with its $\text{PAF}_c = \langle\{a,b\}, \{(a,b)\}, \{\}, \{(a,a),(b,b),(b,a)\}\rangle$ and its $\text{PAF}_o = \langle\{a,b\}, \{(a,b)\}, \{(b,a)\}, \{(a,a),(b,b)\}\rangle$ represented by:



More generally, many other PAFs can be obtained from a given AF, keeping the same set of arguments $\mathcal{A}$, the same attack relation $\mathcal{R}$, and taking $\mathcal{I}$ as a subset of $\{(a,b)|a \neq b \text{ and } (a,b) \notin \mathcal{R}\}$.

4.2 Viewing a PAF as a Set of AFs

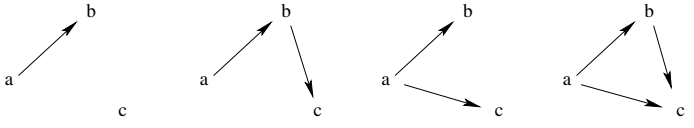
Using the notion of completion, introduced in [1,8], a PAF compactly represents a set of AFs.

[4] In order to avoid the derivability of the arguments which could be self-attacked: in Dung's approach, whatever the semantics, an argument a which is derivable is such that $(a,a) \notin \mathcal{R}$.

Definition 5. *Let* $\mathsf{PAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{I}, \mathcal{N} \rangle$. *Let* $\mathsf{AF} = \langle \mathcal{A}, \mathcal{R}' \rangle$. AF *is a* completion *of* PAF *if and only if* $\mathcal{R} \subseteq \mathcal{R}' \subseteq \mathcal{R} \cup \mathcal{I}$. *The set of all completions of* PAF *is denoted by* $\mathcal{C}(\mathsf{PAF})$.

Note that if $\mathcal{I}$ is the empty set, the PAF has only one completion. This is the case for instance when PAF is the closed PAF associated with $\mathsf{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$. AF is the unique completion of its closed PAF.

Example 3. Consider $\mathsf{PAF} = \langle \mathcal{A} = \{a, b, c\}, \mathcal{R} = \{(a, b)\}, \mathcal{I} = \{(a, c), (b, c)\}, \mathcal{A} \times \mathcal{A} \setminus (\mathcal{R} \cup \mathcal{I}) \rangle$. PAF has the following completions:



5 Semantics for PAF

As for the AFs, we are interested in a collective acceptability. So, we have to restate the notions of conflict-free set and acceptable set in the context of PAF.
Let $\mathsf{PAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{I}, \mathcal{N} \rangle$. Let $S \subseteq \mathcal{A}$.

Definition 6. S *is* $\mathcal{R}$-conflict-free *for* PAF *iff there exist no* a_i, a_j *in* S *such that* $a_i \mathcal{R} a_j$.
S *is* $\mathcal{RI}$-conflict-free *for* PAF *iff there exist no* a_i, a_j *in* S *such that* $a_i \mathcal{R} a_j$ *or* $a_i \mathcal{I} a_j$.

The $\mathcal{R}$-conflict-free notion corresponds exactly to Dung's conflict-free notion (see Property 3). The $\mathcal{RI}$-conflict-free notion is a more cautious one. If S is $\mathcal{RI}$-conflict-free, we are sure that for any pair (a, b) of arguments in S, a does not attack b.

Definition 7. *An argument* a *is* $\mathcal{R}$*-acceptable w.r.t.* S *for* PAF *iff* $\forall b \in \mathcal{A}$ *such that* $b \mathcal{R} a$, $\exists c \in S$ *such that* $c \mathcal{R} b$. S *is* $\mathcal{R}$-acceptable *for* PAF *iff* $\forall a \in S$, a *is* $\mathcal{R}$*-acceptable w.r.t.* S *for* PAF.

An argument a *is* $\mathcal{RI}$*-acceptable w.r.t.* S *for* PAF *iff* $\forall b \in \mathcal{A}$ *such that* $b \mathcal{R} a$ *or* $b \mathcal{I} a$, $\exists c \in S$ *such that* $c \mathcal{R} b$. S *is* $\mathcal{RI}$-acceptable *for* PAF *iff* $\forall a \in S$, a *is* $\mathcal{RI}$*-acceptable w.r.t.* S *for* PAF.

The $\mathcal{R}$-acceptability corresponds exactly to Dung's acceptability (see Property 3). The $\mathcal{RI}$-acceptability is a very cautious acceptability: in order to accept a, first we consider not only known attacks on a but also potential attacks on a, and secondly S must defend a only with certainly known attacks.

Following Dung's method, several *semantics for admissibility* can be defined. In particular such semantics can be obtained by the combination of the above defined notions of conflict-free and acceptability. However, among the four possible combinations, two of them are equivalent because of the following property:

Property 1. If S is $\mathcal{R}$-conflict-free and $\mathcal{RI}$-acceptable for PAF then S is $\mathcal{RI}$-conflict-free for PAF.

Proof. Consider $(a, b) \in S \times S$ such that $a\mathcal{I}b$. S is $\mathcal{RI}$-acceptable so $\exists c \in S$ such that $c\mathcal{R}a$ which is contradictory with S $\mathcal{R}$-conflict-free. So S is also $\mathcal{RI}$-conflict-free.

Definition 8. *S is an* admissible set *for* PAF *iff S is $\mathcal{R}$-conflict-free and $\mathcal{R}$-acceptable for* PAF.

*S is a $\mathcal{R}$-*admissible set *for* PAF *iff S is $\mathcal{RI}$-conflict-free and $\mathcal{R}$-acceptable for* PAF.

*S is a $\mathcal{RI}$-*admissible set *for* PAF *iff S is $\mathcal{RI}$-conflict-free and $\mathcal{RI}$-acceptable for* PAF.

Note that the first admissibility corresponds exactly to Dung's admissibility (see Property 3). And the following interesting property holds:

Property 2. S is $\mathcal{RI}$-admissible for PAF $\Rightarrow$ S is $\mathcal{R}$-admissible for PAF $\Rightarrow$ S is admissible for PAF.

The proof is obvious. Due to Definitions 6, 7, 8.

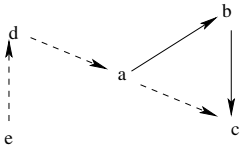
Then using the maximality for $\subseteq$, three kinds of preferred extensions can be defined.

Definition 9. *S is a* preferred extension *of* PAF *iff S is maximal for $\subseteq$ among the admissible sets for* PAF.

*S is a $\mathcal{R}$-*preferred extension *of* PAF *iff S is maximal for $\subseteq$ among the $\mathcal{R}$-admissible sets for* PAF.

*S is a $\mathcal{RI}$-*preferred extension *of* PAF *iff S is maximal for $\subseteq$ among the $\mathcal{RI}$-admissible sets for* PAF.

Example 4. Consider the PAF represented by the following figure:



$\{a, c\}$ is $\mathcal{R}$-conflict-free but it is not $\mathcal{RI}$-conflict-free.
c is $\mathcal{R}$-acceptable w.r.t. $\{a\}$ but it is not $\mathcal{RI}$-acceptable w.r.t. $\{d\}$.
$\{e, d, a, c\}$ is a preferred extension but it is not a $\mathcal{R}$-preferred extension or a $\mathcal{RI}$-preferred extension (because it is not $\mathcal{RI}$-conflict-free).
$\{e, a\}$ is a $\mathcal{R}$-preferred extension but it is not a $\mathcal{RI}$-preferred extension. $\{e\}$ is the $\mathcal{RI}$-preferred extension.

6 Properties

The first properties are about the inclusion links between the different types of extensions. Then, some properties about complexity of the derivability process are given.

6.1 Inclusion Links

The main property explicits the links between the less cautious semantics for a given PAF $= \langle \mathcal{A}, \mathcal{R}, \mathcal{I}, \mathcal{N} \rangle$ and Dung's semantics for the associated AF $= \langle \mathcal{A}, \mathcal{R} \rangle$ (same set of arguments and same attack relation). It also gives the particular links between the semantics for PAF and Dung's semantics for the associated AF when PAF is the closed PAF of AF:

Property 3.

1. The conflict-free sets for AF are exactly the $\mathcal{R}$-conflict-free sets for PAF;
2. Let $S \subseteq \mathcal{A}$, a is acceptable w.r.t S for AF iff a is $\mathcal{R}$-acceptable w.r.t. S for PAF;
3. The admissible sets for AF are exactly the admissible sets for PAF;
4. The preferred extensions of AF are exactly the preferred extensions of PAF.

In the particular case where $\mathcal{I} = \varnothing$ (for instance, if PAF is the closed PAF of AF):

1. The conflict-free sets for AF are exactly the $\mathcal{RI}$-conflict-free sets for PAF;
2. Let $S \subseteq \mathcal{A}$, a is acceptable w.r.t S for AF iff a is $\mathcal{RI}$-acceptable w.r.t. S for PAF;
3. The admissible sets for AF are exactly the $\mathcal{R}$-admissible sets for PAF and the $\mathcal{RI}$-admissible sets for PAF ;
4. The preferred extensions of AF are exactly the $\mathcal{R}$-preferred extensions of PAF and the $\mathcal{RI}$-preferred extensions of PAF.

The proof is obvious due to Definitions 6, 7, 8 and 9 and to the fact that PAF and AF use the same set of arguments and the same attack relation. And for the particular case, the proof also uses the fact that $\mathcal{I} = \varnothing$ when PAF is the closed PAF of AF.

The following properties show the inclusion links between semantics for PAF.

Property 4.

1. The set of all admissible (resp. $\mathcal{R}$-admissible, $\mathcal{RI}$-admissible) sets for PAF is a complete partially ordered set in $(2^{\mathcal{A}}, \subseteq)$.
2. For every admissible (resp. $\mathcal{R}$-admissible, $\mathcal{RI}$-admissible) set S for PAF, there exists at least one preferred extension (resp. $\mathcal{R}$-preferred extension, $\mathcal{RI}$-preferred extension) E of PAF s.t. $S \subseteq E$.
3. There always exists at least one preferred extension (resp. $\mathcal{R}$-preferred extension, $\mathcal{RI}$-preferred extension) of PAF.

Proof.
The case concerning the admissible sets and the preferred extensions is obtained directly of [7] and Property 3. The case concerning the $\mathcal{R}$-admissible sets and the $\mathcal{R}$-preferred extensions is given by (the proof is the same type for the $\mathcal{RI}$-admissible sets and the $\mathcal{RI}$-preferred extensions):

1. For proving that a set S is a complete partially ordered set (cpo), we must to prove that (i) S has a least element, and that (ii) each directed subset of S has a least upper bound in S.
   For the point (i), it is easy to see that the set of the $\mathcal{R}$-admissible sets has a least element w.r.t. $\subseteq$ since $\varnothing$ is always $\mathcal{R}$-admissible for PAF.
   For the point (ii), let $X = \{S_1, \ldots, S_n\}$ be a directed set of $\mathcal{R}$-admissible sets for PAF. Hence, every S_j is $\mathcal{R}$-admissible and for all S_j and S_k, there exists a $\mathcal{R}$-admissible set S such that $S_j \subseteq S$ and $S_k \subseteq S$. Clearly, if the upper bound $\bigvee_X$ of X exists, then $\bigcup_j S_j \leq \bigvee_X$ (because the order considered is $\subseteq$). Hence, $\bigcup_j S_j = \bigvee_X$ and $\bigvee_X$ exists, if $\bigcup_j S_j$ is $\mathcal{R}$-admissible.
   Proof that $\bigcup_j S_j$ is $\mathcal{R}$-admissible. Every S_j is $\mathcal{R}$-admissible, hence if $a \in S_j$, a is $\mathcal{R}$-acceptable w.r.t. S_j and then a is $\mathcal{R}$-acceptable w.r.t. $\bigcup_j S_j$.
   Now, assume that $\bigcup_j S_j$ is not $\mathcal{RI}$-conflict-free. Then, there exists $a \in S_i$ and $b \in S_j$ such that $a\mathcal{R}b$ or $a\mathcal{I}b$. In other words, $S_i \cup S_j$ is not $\mathcal{RI}$-conflict-free. But, as X is a directed set, there exists $S \supseteq S_i \cup S_j$ such that S is $\mathcal{R}$-admissible, hence $\mathcal{RI}$-conflict-free. This contradicts the fact that $a\mathcal{R}b$ or $a\mathcal{I}b$. Hence, $\bigcup_j S_j$ is $\mathcal{RI}$-conflict-free.

2. Direct of the previous point (take X the set of all $\mathcal{R}$-admissible sets containing S; $\bigvee_X$ is a $\mathcal{R}$-preferred subset containing S).
3. Obvious because $\varnothing$ is a $\mathcal{R}$-admissible set for PAF.

The following property shows the cautiousness of the $\mathcal{RI}$-preferred semantics w.r.t. the $\mathcal{R}$-preferred semantics and the cautiousness of the $\mathcal{R}$-preferred semantics w.r.t. the preferred semantics in a PAF.

Property 5. Let E be a $\mathcal{RI}$-preferred extension (resp. $\mathcal{R}$-preferred extension) of PAF. There always exists at least one $\mathcal{R}$-preferred extension (resp. preferred extension) E' of PAF such that $E \subseteq E'$.

Proof. Obvious because $\mathcal{RI}$-admissibility for PAF implies $\mathcal{R}$-admissibility for PAF which implies admissibility for PAF and, following Property 4, a $\mathcal{RI}$-preferred extension of PAF is included in at least one $\mathcal{R}$-preferred extension of PAF which is included in at least one preferred extension of PAF.

A consequence of the previous property (and of Property 3) is the fact that every $\mathcal{R}$-preferred extension (resp. $\mathcal{RI}$-preferred extension) of $\mathsf{PAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{I}, \mathcal{N} \rangle$ is included in at least one preferred extension of $\mathsf{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$. So, our semantics is more constrained than Dung's semantics.

The following property shows that an argument which is certainly unattacked is always in an extension of the PAF:

Property 6. Let $a \in \mathcal{A}$ such that $\nexists b \in \mathcal{A}$ and ($b\mathcal{I}a$ or $b\mathcal{R}a$).

1. a belongs to each preferred extension of PAF;
2. a belongs to at least one $\mathcal{R}$-preferred extension (but not always to each $\mathcal{R}$-preferred extension) of PAF;
3. a belongs to each $\mathcal{RI}$-preferred extension of PAF.

Proof. The first point is obvious because a is non-attacked so we can use Dung's results.

The second point is given by: $\{a\}$ is $\mathcal{RI}$-conflict-free and $\mathcal{R}$-acceptable. So $\{a\}$ is $\mathcal{R}$-admissible and is included in at least one $\mathcal{R}$-preferred extension.

However, the PAF a - - - - - - - - ➤ b ——————➤ c has two $\mathcal{R}$-preferred extensions $\{a\}$ and $\{b\}$ and a does not belong to each one.

For the last point, let E be a $\mathcal{RI}$-preferred extension. If $a \notin E$, consider $E \cup \{a\}$, which strictly contains E. $E \cup \{a\}$ is $\mathcal{RI}$-conflict-free and also $\mathcal{RI}$-acceptable since no argument in $\mathcal{A}$ attacks or ignores a. That contradicts the fact that E is a $\subseteq$-maximal $\mathcal{RI}$-admissible set.

The next properties give some information about handling the cycles of attack or ignorance[5] in a PAF.

[5] A cycle $\{a_0, \ldots, a_{n-1}\}$ of attack or ignorance is a set of n arguments defined by: $\forall i = 0 \ldots n-1$, $a_i \mathcal{R} a_{(i+1) \text{modulo } n}$ or $a_i \mathcal{I} a_{(i+1) \text{modulo } n}$, and $\nexists T \subseteq \{a_0, \ldots, a_{n-1}\}$ such that T is a cycle; so our cycle is always elementary (it does not contain 2 edges with the same initial extremity or the same ending extremity). n is the length of the cycle.

Property 7. Let $a \in \mathcal{A}$ such that $a\mathcal{I}a$. a belongs neither to a $\mathcal{R}$-preferred extension nor to a $\mathcal{RI}$-preferred extension of PAF (but a may belong to a preferred extension of PAF).

Let $a \in \mathcal{A}$ such that $a\mathcal{R}a$. There is no preferred extension (resp. $\mathcal{R}$-preferred extension, $\mathcal{RI}$-preferred extension) of PAF which contains a.

The proof is obvious due to the fact that $\{a\}$ is not $\mathcal{RI}$-conflict-free when $a\mathcal{I}a$, and is not $\mathcal{R}$-conflict-free when $a\mathcal{R}a$.

Property 8. Let $\{a_0, \ldots, a_{n-1}\}$ be an odd-length cycle of attack or ignorance which is certainly unattacked[6]. There is no $\mathcal{RI}$-admissible set for PAF which can contain an element of the cycle.

Proof. The proof is realized by *reductio ad absurdum*. Assume that $\exists E$ $\mathcal{RI}$-admissible set and $\exists a_i$ on the cycle such that $a_i \in E$. If $a_i \in E$ then a_i is $\mathcal{RI}$-acceptable w.r.t. E. However, a_i belongs to the cycle, so a_i is certainly not unattacked and there must have in E arguments for defending it. Consider a new numbering of the arguments of the cycle defined by: a'_0 is $a_{(i+1) \text{modulo } n}$, ..., a'_{n-1} is a_i. With this numbering, one can identified the set of the arguments of the cycle (they must be on the cycle due to the assumption that the cycle is certainly unattacked) which can be mandatory for making a_i $\mathcal{RI}$-acceptable w.r.t. E: $\{a'_{n-3}, \ldots, a'_2, a'_0\}$. So, a_i and $a_{(i+1) \text{modulo } n}$ must belong to E if a_i is $\mathcal{RI}$-acceptable w.r.t. E. This is contradictory with the fact that E is $\mathcal{RI}$-conflict-free, so the initial assumption is false and $a_i \notin E$.

This property does not hold in the case of even cycles. Thus, our approach departs from [10] who consider that odd-length and even-length cycles in an argumentation framework should be considered in the same way. We think that our approach is more cautious.

Note that even if there is no cycle in a PAF, one can have several $\mathcal{R}$-preferred extensions of this PAF (which is not possible with Dung's approach). For instance, consider the PAF a - - - - - - - - ➤ b ——————➤ c, with the two $\mathcal{R}$-preferred extensions $\{a\}$ and $\{b\}$.

The last properties of this section give some links between a PAF and its completions or between an AF and its closed or open PAFs.

Property 9. Let $S \subseteq \mathcal{A}$. S is $\mathcal{RI}$-conflict-free for PAF iff S is conflict-free for any completion of PAF.

Proof. $\Rightarrow$ Assume there exists a completion $\mathsf{AF}' = \langle \mathcal{A}, \mathcal{R}' \rangle$ of PAF such that S is not conflict-free for AF'. There exists a pair (a, b) of arguments in S such that $a\mathcal{R}'b$. By definition of a completion, $\mathcal{R}' \subseteq \mathcal{R} \cup \mathcal{I}$, so S is not $\mathcal{RI}$-conflict-free for PAF.

$\Leftarrow$ If S is conflict-free for any completion of PAF, then S is conflict-free for the completion $\langle \mathcal{A}, \mathcal{R} \cup \mathcal{I} \rangle$, which means exactly that S is $\mathcal{RI}$-conflict-free for PAF.

We also have a link between the conflict-free property for an AF and the conflict-free property for its open or closed PAF:

Property 10. Let $S \subseteq \mathcal{A}$. If S is conflict-free (in Dung's sense) for AF, then S is $\mathcal{R}$-conflict-free (resp. $\mathcal{RI}$-conflict-free) for the open (resp. closed) PAF associated with AF.

The proof is obvious.

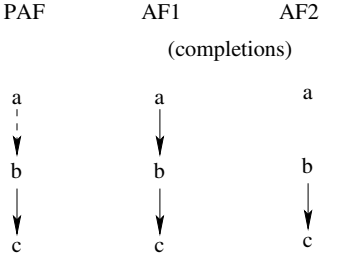
[6] *i.e.* such that there do not exist $x \in \mathcal{A} \setminus \{a_0, \ldots, a_{n-1}\}$ and $0 \leq i \leq n-1$ such that $x\mathcal{R}a_i$ or $x\mathcal{I}a_i$.

Property 11. Let $S \subseteq \mathcal{A}$. S is $\mathcal{RI}$-admissible for PAF $\Rightarrow$ S is admissible for each completion of PAF.

The proof follows Property 9 and uses the fact that if a is $\mathcal{RI}$-acceptable w.r.t. S for PAF then a is acceptable w.r.t. S for $\langle \mathcal{A}, \mathcal{R}' \rangle$ with $\mathcal{R} \subseteq \mathcal{R}' \subseteq \mathcal{R} \cup \mathcal{I}$.

Property 12. Let $\mathsf{AF}' = \langle \mathcal{A}, \mathcal{R}' \rangle$ a completion of PAF. Let E be a preferred extension of AF'. The fact that $a \in E$ does not imply that there exists a $\mathcal{R}$-preferred extension or a $\mathcal{RI}$-preferred extension of PAF which contains a.

Proof. Consider PAF represented by:



The $\mathcal{R}$-preferred extensions of PAF are $\{a\}$ and $\{b\}$.
The $\mathcal{RI}$-preferred extension of PAF is $\{a\}$.
The preferred extension of AF_1 is $\{a, c\}$ and the preferred extension of AF_2 is $\{a, b\}$.
c belongs to the extension of AF_1 but it belongs neither to a $\mathcal{R}$-preferred extension of PAF, nor to the $\mathcal{RI}$-preferred extension of PAF.

Property 13. If E is a $\mathcal{RI}$-preferred extension of PAF then for each completion AF_i of PAF, $i = 1 \ldots n$, there exists a preferred extension of AF_i denoted by E_i such that $E \subseteq E_1 \cap \ldots \cap E_n$.

Proof. E is $\mathcal{RI}$-admissible for PAF. From Property 11, E is admissible in each completion AF_i of PAF. So, from [7], for each AF_i, there exists a preferred extension E_i of AF_i containing E. So, $E \subseteq E_1 \cap \ldots \cap E_n$.

6.2 Complexity Results

Now, let us consider some complexity issues. Indeed, in an AI perspective, it is important to determine how hard are the new inference relations we pointed out w.r.t. the computational point of view. We assume the reader acquainted with basic notions of complexity theory, especially the complexity classes P, NP, coNP and the polynomial hierarchy (see e.g. [11]), and with complexity results in Dung's framework (see [12]).

[13] have shown that considering sets of arguments (instead of single arguments) as input queries for the inference problem[7] does not lead to a complexity shift when Dung's inference relations are considered. As to inference relations concerning the PAFs, the same conclusion can be drawn.

First of all, it is easy to show that, given a finite partial argumentation framework PAF, deciding whether a given argument $\mathcal{R}$-interacts or $\mathcal{I}$-interacts with a givenargument is in P, and deciding whether a set of arguments is $\mathcal{R}$-conflict-free for PAF and deciding whether a set of arguments is $\mathcal{RI}$-conflict-free for PAF are in P. Accordingly, deciding whether a given set of arguments is admissible for PAF, deciding whether a

[7] The purpose is to determine whether such sets are derivable from a given finite argumentation framework AF.

given set of arguments is $\mathcal{R}$-admissible for PAF and deciding whether a given set of arguments is $\mathcal{RI}$-admissible for PAF are in P.

Besides, deciding whether a set of arguments S is a preferred extension of PAF, deciding whether a set of arguments S is a $\mathcal{R}$-preferred extension of PAF and deciding whether a set of arguments S is a $\mathcal{RI}$-preferred extension of PAF are in coNP (in order to show that the complementary problem is in NP, it is sufficient to guess a proper superset S' of S and to check in polynomial time that S' is admissible for PAF, $\mathcal{R}$-admissible for PAF or $\mathcal{RI}$-admissible for PAF).

Property 14. Let $\mathsf{PAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{I}, \mathcal{N} \rangle$ be a finite partial argumentation framework and $S \subseteq \mathcal{A}$.

1. Deciding whether S is included in a preferred extension of PAF (resp. in a $\mathcal{R}$-preferred extension of PAF, in a $\mathcal{RI}$-preferred extension of PAF) is NP-complete;
2. Deciding whether S is included in all preferred extensions of PAF (resp. in all $\mathcal{R}$-preferred extensions of PAF, in all $\mathcal{RI}$-preferred extensions of PAF) is Π_2^p-complete.

Proof. `For the first problem:` *Membership:* Deciding whether a given set of arguments S is included in a preferred extension of PAF is in NP (according to Property 4, it is sufficient to guess a set $E \subseteq \mathcal{A}$ and to check in polynomial time if E is an admissible set for PAF and to check that S is included in E). *Hardness:* We build in polynomial time a polynomial functional reduction f from the problem to decide whether a given set of arguments S is included in a preferred extension of $\mathsf{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$. Let $\mathsf{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$ be a finite argumentation framework and $S \subseteq \mathcal{A}$. $f : \langle \mathsf{AF} = \langle \mathcal{A}, \mathcal{R} \rangle, S \subseteq \mathcal{A} \rangle \mapsto \langle \mathsf{PAF}_c = \langle \mathcal{A}, \mathcal{R}, \varnothing, (\mathcal{A} \times \mathcal{A}) \setminus \mathcal{R} \rangle, S' \subseteq \mathcal{A} \rangle$ with $S' = S$. According to Property 3, S is included in a preferred extension of AF iff S' is included in a preferred extension of PAF_c. As, according to [13], deciding whether a given set of arguments S is included in a preferred extension of AF is in NP-complete, deciding whether a given set of arguments S is included in a preferred extension of AF is in NP-complete.

`For the second problem:` *Membership:* Deciding whether a given set of arguments S is included in every preferred extension of PAF is in Π_2^p (in order to show that the complementary problem is in Σ_2^p, it is sufficient to guess a set $E \subseteq \mathcal{A}$ and to check in polynomial time using an NP oracle that E is a preferred extension of AF and that S is not included in E). *Hardness:* We build in polynomial time a polynomial functional reduction f from the problem to decide whether a given set of arguments S is included in all preferred extensions of $\mathsf{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$. Let $\mathsf{AF} = \langle \mathcal{A}, \mathcal{R} \rangle$ be a finite argumentation framework and $S \subseteq \mathcal{A}$. $f : \langle \mathsf{AF} = \langle \mathcal{A}, \mathcal{R} \rangle, S \subseteq \mathcal{A} \rangle \mapsto \langle \mathsf{PAF}_c = \langle \mathcal{A}, \mathcal{R}, \varnothing, (\mathcal{A} \times \mathcal{A}) \setminus \mathcal{R} \rangle, S' \subseteq \mathcal{A} \rangle$ with $S' = S$. According to Property 3, S is included in all preferred extensions of AF iff S' is included in all preferred extensions of PAF_c. As, according to [13], deciding whether a given set of arguments S is included in all preferred extensions of AF is in Π_2^p-complete, deciding whether a given set of arguments S is included in all preferred extensions of AF is in Π_2^p-complete.

The proof is similar for the $\mathcal{R}$-preferred extensions and the $\mathcal{RI}$-preferred extensions.

7 Conclusion

In this paper, we have studied an extension of Dung's argumentation framework (AF) introduced in [1,8]. This extension, called "partial argumentation framework" (PAF), consists in introducing another interaction between arguments which represents some ignorance about the existence of an attack between two arguments. In [1,8], the PAFs

have been used in order to merge different AFs. Here, we have proposed a more formal study of acceptability semantics in a PAF.

So we have generalized the basic notions proposed by Dung (conflict-free set, acceptability of an argument, admissibility) to the PAF and we have identified several interesting semantics.

The main properties of these semantics concern the inclusion links between them; a taxonomy of these semantics can be found where the more general case is given by Dung's semantics which take into account only the attacks, and the less general case is given by the semantics, proposed in this paper, which take into account attacks *and* ignorance in the same time in the notion of conflict-free *and* in the notion of acceptability of an argument.

Another fundamental property concerns the complexity of the problems related to our new semantics: the generalisation to a PAF does not imply an increasing cost of the temporal complexity in the worst case (the complexity results are exactly the same as those given for the AF).

Future works concern the development of algorithms in order to compute efficiently the extensions of our semantics. A more theoretical issue is to apply our semantics directly on the merging process proposed in [1,8] in order to reduce the cost of this merging.

References

1. Coste-Marquis, S., Devred, C., Konieczny, S., Lagasquie-Schiex, M.C., Marquis, P.: Argumentation and fusion. In: Proc. of the 20^{th} AAAI, Pittsburg, Pennsylvania (2005)
2. Krause, P., Ambler, S., Elvang, M., Fox, J.: A logic of argumentation for reasoning under uncertainty. Computational Intelligence 11 (1), 113–131 (1995)
3. Prakken, H., Vreeswijk, G.: Logics for defeasible argumentation. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic, vol. 4, pp. 218–319. Kluwer Academic Publishers, Dordrecht (2002)
4. Bondarenko, A., Dung, P., Kowalski, R., Toni, F.: An abstract, argumentation-theoretic approach to default reasoning. Artificial Intelligence 93, 63–101 (1997)
5. Amgoud, L., Maudet, N., Parsons, S.: Modelling dialogues using argumentation. In: Fourth International Conference on MultiAgent Systems (ICMAS'2000), Boston, MA, pp. 31–38 (July 2000)
6. Chesñevar, C.I., Maguitman, A.G., Loui, R.P.: Logical models of argument. ACM Computing surveys 32(4), 337–383 (2000)
7. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77, 321–357 (1995)
8. Devred, C.: Contribution à l'étude des inférences argumentatives: spécialisations et généralisations du cadre de Dung. PhD thesis, Université d'Artois (2006)
9. Hunter, A.: Approximate arguments for efficiency in logical argumentation. In: Proceedings of the International workshop in Non-monotonic Reasoning (NMR) (2006)
10. Baroni, P., Giacomin, M., Guida, G.: Scc-recursiveness: a general schema for argumentation semantics. Artifical Intelligence 168, 162–210 (2005)
11. Papadimitriou, C.: Computational complexity. Addison-Wesley, Reading (1994)
12. Dunne, P.E., Bench-Capon, T.J.: Coherence in finite argument system. Artificial Intelligence 141(1-2), 187–203 (2002)
13. Coste-Marquis, S., Devred, C., Marquis, P.: Symmetric argumentation frameworks. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 317–328. Springer, Heidelberg (2005)

Dialectical Proof Theories for the Credulous Prudent Preferred Semantics of Argumentation

Caroline Devred[1] and Sylvie Doutre[2]

[1] CRIL-CNRS – Université d'Artois
devred@cril.univ-artois.fr
LERIA – Université d'Angers
caroline.devred@info.univ-angers.fr
[2] IRIT – Université Toulouse 1
doutre@irit.fr

Abstract. In Dung's argumentation system, acceptable sets of arguments are defined as sets of arguments that attack all their attackers, and that do not contain any direct contradiction. However, in many applications, the presence of indirect contradictions should prevent a set from being acceptable. The family of prudent semantics has been proposed as an answer to this problem. We are interested in this paper in determining whether a given set of arguments is included in at least one acceptable set under the prudent preferred semantics. To this end, we propose a dialectical framework and several proof theories.

1 Introduction

Argumentation is a general approach for nonmonotonic reasoning, whose main feature is the use of arguments to draw conclusions based on the way arguments interact ([1,2]). Argumentation has applications in various domains such as law, medicine, e-democracy. At a high level of abstraction, argumentation systems can be represented by a set of abstract arguments (whose internal structure and origin is unknown), and a binary relation between these arguments describing how they contradict one another. Such a representation, proposed by Dung in [3], is powerful enough to encompass several problems related to nonmonotonic reasoning and logic programming. Dung's framework has been refined and extended by several authors (e.g. [4,5,6,7,8]).

One of the most important questions concerning abstract argumentation systems is to define which arguments are acceptable. The common ground of Dung [3]'s acceptability semantics is that a set of arguments is acceptable if it attacks all its attackers and if it does not contain two arguments such that one *directly* attacks the other. However, in an application domain such as law, it may not be suitable, and even not prudent, that an acceptable set of arguments contains indirect contradictions. For example, consider the case of a barrister who wants to prove the innocence of her client; the set of arguments she will present to the court should not contain any argument that casts a doubt on her defence, and an argument acceptable in Dung's sense, but that indirectly attacks an argument of the defence, may cast a doubt. [7] presents another example that illustrates that indirect contradictions are not suitable either in the context of a political discourse.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 271–282, 2007.

As an answer to this problem, Devred *et al.* [6] proposed a refinement of Dung's semantics that avoids indirect contradictions: the *prudent* semantics. Acceptable sets of arguments under these semantics are called p-extensions. In this paper, we are interested in the prudent semantics that defines preferred p-extensions. This semantics is adapted from a widespread Dung's semantics, the preferred semantics.

A question frequently addressed for Dung's argumentation framework is: is a given argument in some/all extensions of an argumentation system? This allows to know the credulously/sceptically accepted arguments. Several works, e.g. [9,10,11,12], give answers to this question for Dung's preferred semantics, [13] gives an answer for Dung's grounded semantics; these answers are presented in the form of dialogues, or of sets of dialogues, which explain the answer given. Such dialogues differ from dialogues such as the ones by [14,15], since they do not aim at resolving a conflict of opinion between participants, by making or conceding claims, and asking for or providing reasons for the claims; their goal is only to prove the acceptability of an argument.

The question of the acceptability of an argument is in fact a particular case of a more general question, that is: is a given *set* of arguments included in some/all extensions of an argumentation system? In other words, is a set of arguments credulously/sceptically accepted?

The acceptability of an argument comes down to checking the acceptability of the set that contains this single argument. However, notice that the problem of the credulous acceptance of a set of arguments does not boil down to the credulous acceptance of each of its arguments. Actually, two arguments may be acceptable because each of them belongs to an acceptable set of argument, but, because they may contradict each other, they do not belong to a same acceptable set of arguments, and thus the set that contains these two arguments is not included in any extension.

Our goal in this paper is to give an answer to the credulous acceptance problem for a set of arguments under the prudent preferred semantics. [10]'s dialectical framework has been designed to draw proofs for the acceptability of an argument, but it can be easily extended to the acceptability of a set of arguments, as it has already been partially done by [11]. So we first present an extended framework, and we then adapt [10]'s proof theories for an argument under the preferred semantics to a set of arguments under the prudent preferred semantics.

The paper is built as follows: Dung's framework and the prudent semantics are presented in Section 2. Section 3 presents the extension of [10]'s dialectical framework to a set of arguments. Section 4 presents dialectical proof theories for the acceptability of a set of arguments under the prudent preferred semantics. We conclude in Section 5.

2 Argumentation Framework and Semantics

This section briefly presents Dung's argumentation framework [3], the preferred semantics, and Devred *et al.* prudent preferred semantics [6].

Definition 1. *[3] An* argumentation framework *is a pair* $\langle A, R \rangle$ *where* A *is a finite set of so-called* arguments *and* R *is a binary relation over* A *(*$R \subseteq A \times A$*). Given two arguments* a *and* b*,* $(a, b) \in R$ *means that* a attacks b *(*a *is said to be an* attacker *of* b*).*

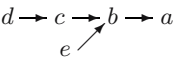


Fig. 1. Argumentation framework AF_1

An argumentation framework is nothing but a directed graph, whose vertices are the arguments and edges correspond to the elements of R.

Example 1. Let $AF_1 = \langle A, R\rangle$ be an argumentation framework with $A = \{a, b, c, d, e\}$ and $R = \{(b, a), (e, b), (c, b), (d, c)\}$. The graph for AF_1 is depicted on Figure 1.

In the following definitions and notations, we assume that an argumentation framework $\langle A, R\rangle$ is given.

Dung's semantics rely upon the notion of defence[1] defined as follows:

Definition 2. *[3] An argument $a \in A$ is* defended *by a set $S \subseteq A$ if and only if for every $b \in A$ that attacks a, there exists $c \in S$ that attacks b.*

Based on defence and on the rejection of direct attacks between arguments, Dung defines admissible sets and preferred extensions.

Definition 3. *[3] Let $S \subseteq A$ be a set of arguments.*

- *S is* conflict-free *if and only if for every $a, b \in S$, we have $(a, b) \notin R$.*
- *S is* admissible *if and only if S is conflict-free and every argument in S is defended by S.*
- *S is a* preferred extension *if and only if it is maximal w.r.t. $\subseteq$ among the set of admissible sets.*

Example 2 (Example 1 (contd)). $\{a, d\}$ is conflict-free. a is defended by $\{e\}$. $\{a, d, e\}$, $\{a, e\}$, $\{d, e\}$, $\{e\}$, $\{d\}$ and $\emptyset$ are the admissible sets. $\{a, d, e\}$ is the unique preferred extension of AF.

In this example, argument d does not *directly* attack a, but it attacks an attacker of an attacker of a: the attack of d on a is *indirect*. Even though a is defended by $\{e\}$, the presence of d in the preferred extension casts a doubt on the acceptability of a. For applications where this may be a problem (e.g. political discourse, lawyer's defence), the definition of acceptability should be refined in order for an acceptable set of argument to be without any indirect attack. This is what Devred *et al.* [6] did by defining prudent semantics.

Definition 4. *[6] Let $a, b \in A$ be two arguments. a* indirectly attacks *b if and only if there exists an odd-length path from a to b.*

Definition 5. *[6] Let $S \subseteq A$ be a set of arguments.*

- *S is* without indirect conflict *if and only if there is no pair of arguments a and b of S such that a indirectly attacks b.*

[1] This notion of defence is originally defined by Dung as "acceptability with respect to a set".

- S *is* p-admissible *if and only if every* $a \in S$ *is defended by* S *and* S *is without indirect conflict.*
- S *is a* preferred p-extension *if and only if* S *is p-admissible, and* S *is maximal w.r.t.* $\subseteq$ *among the set of p-admissible sets.*

Notice that, since the empty set is always p-admissible, every argumentation framework has at least one preferred p-extension. Moreover, any argument that is part of an odd-length cycle indirectly attacks itself: therefore, such an argument never belongs to a p-admissible set. Devred *et al.* [6] also showed that every p-admissible set is included in at least one preferred p-extension, and that every preferred p-extension is included in a preferred extension.

Example 3 (Example 1 (contd)). d indirectly attacks a. $\{a, e\}$ is without indirect conflict, but it is not the case for $\{a, d\}$. $\{a, e\}$, $\{d, e\}$, $\{d\}$, $\{e\}$ and $\emptyset$ are the p-admissible sets. $\{a, e\}$ and $\{d, e\}$ are the preferred p-extensions.

This example shows that preferred p-extensions capture well the idea that, if an argument indirectly attacks another, these arguments should not be accepted together in a same set: either the one is accepted, or the other, but not both.

Now, an important issue in argumentation is to be able to decide, given any definition of acceptability, which arguments are acceptable. This is how this issue is formally defined for the prudent preferred semantics:

Definition 6. *Let* $S \subseteq A$ *be a set of arguments.*

- S *is* credulously accepted *under the prudent preferred semantics if and only if* S *is included in at least one preferred p-extension of* $\langle A, R \rangle$.
- S *is* sceptically accepted *under the prudent preferred semantics if and only if* S *is included in every preferred p-extension of* $\langle A, R \rangle$.

We say that an argument $a \in A$ *is credulously (resp. sceptically) accepted under the prudent preferred semantics if and only if the set* $\{a\}$ *is.*

Example 4 (Example 1 (contd)). e is sceptically accepted (and indeed, credulously accepted). a is credulously, but not sceptically, accepted under the prudent preferred semantics. The same holds for d. However, the set $\{a, d\}$ is not included in any preferred p-extension: it is neither sceptically, nor credulously accepted. As regards $\{a, e\}$, it is credulously, but not sceptically, accepted.

3 Dialectical Framework

[10] proposed a dialectical framework to build proofs for the acceptability of an argument under the preferred semantics. An extension of this framework for the acceptability of a set of arguments was presented in [11], but this extension is less general than the one we present now.

Basically, a dialectical proof is formalized by a dialogue between two players, a proponent, PRO, and an opponent, OPP. A dialogue takes place in a given argumentation framework and is governed by rules expressed in a so-called legal-move function.

In the sequel, given a set A, A^* denotes the set of finite sequences of elements from A; given a set $S \subseteq A$, $\#(S)$ denotes the cardinality of S.

Definition 7. *A* dialogue type *is a tuple* (A, R, ϕ) *where* $\langle A, R\rangle$ *is an argumentation framework and* $\phi : A^* \to 2^A$ *is a function called* legal-move function. *A* move *in* A *is a pair* $[P, x]$ *where* $P \in \{\text{PRO}, \text{OPP}\}$ *and* $x \in A$. *For a move* $\mu = [P, x]$, *we use* $\text{pl}(\mu)$ *to denote* P *and* $\arg(\mu)$ *to denote* x. *A* dialogue d for a finite set $S \subseteq A$ *in* (A, R, ϕ) *(or* ϕ*-dialogue for short) is a countable sequence* $\mu_0\mu_1 \ldots$ *of moves in* A *such that:*

1. *the first* $\#(S)$ *moves are played by* PRO *to put forward the elements of* S
2. *the subsequent moves are played alternatively by* OPP *and* PRO
3. $\forall i > \#(S), \arg(\mu_{i+1}) \in \phi(\arg(\mu_0) \ldots \arg(\mu_i))$

If S *contains a single argument* x, *we say that* d is about x *(i.e.* $\arg(\mu_0)$*).*

So OPP and PRO play in turn, after PRO has put forward all the elements of S. The legal-move function defines what moves can be used to continue the dialogue. When the set of arguments returned by the legal-move function is empty, the dialogue cannot be continued.

Notation 1. *Let* $d = \mu_0\mu_1 \ldots \mu_i$ *be a finite* ϕ*-dialogue:*

- μ_i *is denoted by* $\text{last}(d)$*;*
- $\phi(\arg(\mu_0) \ldots \arg(\mu_i))$ *is denoted by* $\phi(d)$*;*
- $\text{length}(d)$ *denotes the number of moves in* d*;*
- $\text{PRO}(d)$ *denotes the set of arguments advanced by* PRO *during* d*;*
- *The extension of* d *with a move* μ *in* A *such that* $\mu_0\mu_1 \ldots \mu_i\mu$ *is a* ϕ*-dialogue is denoted by the juxtaposition* $d.\mu$*.*

Definition 8. *The sequence* β *is a* prefix *of the sequence* α *or* α *is an* extension *of* β *if and only if there exists a sequence* γ *such that* α *is obtained by the concatenation of* β *and* γ*, that is,* $\alpha = \beta.\gamma$*.*

We define two winning criteria:

Definition 9. *Given a dialogue type* (A, R, ϕ)*:*

- *A* ϕ*-dialogue* d *is* won by PRO *if and only if* d *is finite, cannot be continued (that is,* $\phi(d) = \emptyset$*), and* $\text{pl}(\text{last}(d)) = \text{PRO}$.
- *A* ϕ-winning strategy *for a set of arguments* S *is a non-empty finite set* D *of finite* ϕ*-dialogues for* S *such that:* $\forall d \in D, \forall d'$ *prefix of* d *such that* $\text{pl}(\text{last}(d')) = \text{PRO}$ *and* $\text{length}(d') \geq \#(S)$, $\forall y \in \phi(d')$, $\exists d'' \in D$ *such that* d'' *is won by* PRO *and* d'' *is an extension of the juxtaposition* $d'.[\text{OPP}, y]$.

In other words, a ϕ-winning strategy for a set of arguments S must show that any ϕ-dialogue about S where the first $\#(S)$ moves have been put forward and where PRO plays the last move, can be extended in a ϕ-dialogue won by PRO whatever the response of OPP to this last move.

Notation 2. *If D is a ϕ-winning strategy for a set of arguments S, we denote by* $\mathrm{PRO}(D)$ *the set* $\cup_{d \in D} \mathrm{PRO}(d)$.

The definition of a dialogue which is won will be used in section 4.1 to define sequential, concise proofs. Winning strategies will be used in section 4.2 to define detailed proofs, that explain clearly each step of the proof.

The following result provides a simpler characterization of ϕ-winning strategies for a set of arguments in which only dialogues which cannot be continued are considered. It is a straightforward extension of [10]'s result to a ϕ-winning strategy for a set of arguments.

Proposition 1. *There exists a ϕ-winning strategy for a set of arguments S if and only if there exists a finite non-empty set D of finite ϕ-dialogues for S won by* PRO *such that:* $\forall d \in D, \forall d'$ *prefix of* d *such that* $\mathrm{last}(d')$ *is played by* PRO *and* $\mathrm{length}(d') \geq \#(S)$, $\forall y \in \phi(d'), \exists d'' \in D$ *such that* d'' *is an extension of the juxtaposition* $d'.[\mathrm{OPP}, y]$.

4 Credulous Acceptance Problem

The credulous acceptance problem for a set of arguments under the prudent preferred semantics is to decide if a given set of arguments S is included in at least one preferred p-extension. The answers that we are going to give for this problem are adapted from the ones by [10,11] for the credulous acceptance problem under the preferred semantics.

First, the answer to the problem is trivial if S is not without indirect conflict: such a set is never included in any p-admissible set. Now, if S is without indirect conflict, the answer should take the form of a proof: if S is credulously accepted, the proof should exhibit a p-admissible set of arguments that contains S; the proof should also exhibit attackers of this set, and the structure of the proof should show how the set defends itself against these attackers (that is, how every argument of the set is defended by the set).

The dialogues introduced in the previous section enable us to distinguish arguments that defend S from those that attack it: the former, as well as the arguments of S itself, are labelled "PRO" in the dialogue, whereas the latter are labelled "OPP". The proof theories that we present below are based on dialogues such that for every attacker, that is, for every move $[\mathrm{OPP}, z]$, there is always a preceding move $[\mathrm{PRO}, y]$ in the sequence, such that z attacks y. This PRO-argument y justifies/explains the presence of z in the proof. Similarly, every defender in a proof, that is, every move of the form $[\mathrm{PRO}, y]$ that is not in the first $\#(S)$ moves, must be immediately preceded in the dialogue by an attacker against which it defends, that is, a move $[\mathrm{OPP}, x]$ such that y attacks x.

Other restrictions can be put on the moves that can appear in a proof of credulous acceptance. Before we present them, we introduce some notations.

Notation 3. *Given an argumentation framework $\langle A, R \rangle$, let $a \in A$ and $S \subseteq A$.*

- $\mathrm{RefI} = \{a \in A \mid a$ *belongs to an odd-length cycle*$\}$
- $R^+(a) = \{b \in A \mid (a, b) \in R\}$;
- $R^-(a) = \{b \in A \mid (b, a) \in R\}$;

– $RI^{\pm}(S) = \{b \in A \mid$ *there exists some* $a \in S$ *such that* a *indirectly attacks* b *or such that* b *indirectly attacks* $a\}$.

Let d be a finite ϕ-dialogue. $RI^{\pm}(\mathrm{PRO}(d))$ contains the arguments which indirectly attack or which are indirectly attacked by an argument advanced by PRO during d. ReflI contains the arguments which belong to an odd-length cycle. If d is to be a proof of acceptance, then PRO attempts to build a p-admissible set of arguments, so PRO cannot choose any argument in $RI^{\pm}(\mathrm{PRO}(d))$ for pursuing the dialogue d, nor any argument in ReflI. Moreover, PRO should not have to repeat itself, that is, it should not have to play an argument that already belongs to $\mathrm{PRO}(d)$. In the sequel, $\mathrm{POSSI}(d)$ denotes the set of arguments which may be chosen by PRO to extend a set (already without indirect conflict) $\mathrm{PRO}(d)$:

$$\mathrm{POSSI}(d) = A \setminus (\mathrm{PRO}(d) \cup RI^{\pm}(\mathrm{PRO}(d)) \cup \mathrm{ReflI}).$$

Note that it is useless for OPP to advance an argument which is attacked by $\mathrm{PRO}(d)$.

4.1 Sequential Proof Theories

The following legal-move function leads to a dialectical proof theory, in which OPP is not obliged to respond to the last argument advanced by PRO. It is adapted from the legal-move function ϕ_1 of [10].

Definition 10. *Given an argumentation framework* $\langle A, R\rangle$*, let* $\phi_{p1} : A^* \rightarrow 2^A$ *be defined by:*

– *if* d *is a dialogue of odd length (next move is by* OPP*),*

$$\phi_{p1}(d) = R^-(\mathrm{PRO}(d)) \setminus R^+(\mathrm{PRO}(d));$$

– *if* d *is a dialogue of even length (next move is by* PRO*),*

$$\phi_{p1}(d) = R^-(\arg(\mathrm{last}(d))) \cap \mathrm{POSSI}(d).$$

Combining a dialogue type (A, R, ϕ_{p1}) and the first winning criterion (dialogue won), we obtain ϕ_{p1}-proofs:

Definition 11. *Let* S *be a non-empty set without indirect conflict. A* ϕ_{p1}*-proof for* S *is a* ϕ_{p1}*-dialogue for* S *won by* PRO*.*

A ϕ_{p1}-proof for a set that contains only one argument x is a proof for the credulous acceptance problem for this argument; we say that it is a ϕ_{p1}-proof for x. In this case, the condition that S has to be without indirect conflict comes down to checking that x does not attack itself.

The following result establishes the soundness and the completeness of the ϕ_{p1}-proof theory. Note that it is only complete for argumentation frameworks for which the set of arguments is finite[2].

[2] This result is an extension of [11]'s Proposition 3 to the prudent preferred semantics.

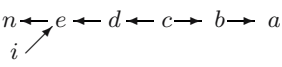


Fig. 2. Argumentation framework AF_2

Proposition 2. *Let $\langle A, R\rangle$ be an argumentation framework, and let $S \subseteq A$. If d is a ϕ_{p1}-proof for S, then $\mathrm{PRO}(d)$ is a p-admissible set containing S. If $S \neq \emptyset$ is in a preferred p-extension of $\langle A, R\rangle$, and if A is finite, then there exists a ϕ_{p1}-proof for S.*

Example 5 (Example 1 (contd)). $d = [\mathrm{PRO}, a].[\mathrm{OPP}, b].[\mathrm{PRO}, c].[\mathrm{OPP}, d]$ is a ϕ_{p1}-dialogue about a lost by PRO. However, the ϕ_{p1}-dialogue $d' = [\mathrm{PRO}, a].[\mathrm{OPP}, b].[\mathrm{PRO}, e]$ is won by PRO. Hence d' is a ϕ_{p1}-proof for a; a is credulously accepted under the prudent preferred semantics.

Example 6. Let $AF_2 = \langle A, R\rangle$ with $A = \{a, b, c, d, e, i, n\}$ and $R = \{(c, b), (b, a), (c, d), (d, e), (e, n), (i, e)\}$. The graph for AF_2 is depicted on Figure 2.

$S_1 = \{a, n\}$ is without indirect conflict. The only ϕ_{p1}-dialogues for S_1 are $d_1 = [\mathrm{PRO}, a].[\mathrm{PRO}, n].[\mathrm{OPP}, b]$, $d_2 = [\mathrm{PRO}, a].[\mathrm{PRO}, n].[\mathrm{OPP}, e].[\mathrm{PRO}, i].[\mathrm{OPP}, b]$ and d'_1 (resp. d'_2) that is identical to d_1 (resp. d_2) except that the first two moves are swapped. In all these dialogues, PRO cannot respond to the argument b played by OPP, since the only attacker of b is c, and c indirectly attacks an argument already put forward by PRO, that is, n. Hence none of these dialogues is won by PRO, so there exists no ϕ_{p1}-proof for S_1. Then S_1 is not credulously accepted under the prudent preferred semantic.

$S_2 = \{a, i\}$ is without indirect conflict. $d_3 = [\mathrm{PRO}, a].[\mathrm{PRO}, i].[\mathrm{OPP}, b].[\mathrm{PRO}, c]$ is a ϕ_{p1}-proof for S_2. Hence S_2 is credulously accepted under the prudent preferred semantic.

Notice that the ϕ_{p1} legal-move function could be refined in order to obtain shorter dialogues. For example, in the case where PRO may have several possible arguments to put forward, PRO should preferably play an argument that is not attacked.

The sequentiality of the ϕ_{p1}-proof theory makes it easy to implement. A drawback is that, in a ϕ_{p1}-dialogue, it is not possible to see which argument is attacked by the argument in a move by OPP: it can be any of the arguments already played by PRO, not necessarily the last one. Consequently, a ϕ_{p1}-proof for a set S does not show how each argument of the resulting p-admissible set is made acceptable, that is, it does not show the *lines of defence* for each argument. This is particularly true for the arguments of S: one cannot see why each of them is on its own acceptable.

This very last point can be solved by defining a proof as a coherent set of ϕ_{p1}-proofs, one for each argument of S.

Definition 12. *Let (A, R, ϕ_{p1}) be a dialogue type, and let $S \subseteq A$. A Φ_{p1}-proof for S is a set E of ϕ_{p1}-proofs such that the following two conditions hold:*

1. *for every $a \in S$, there exists one and only one $d \in E$ such that d is a ϕ_{p1}-proof for a, and*
2. *$\cup_{d \in E}(\mathrm{PRO}(d))$ is without indirect conflict.*

The following result establishes the soundness and the completeness of the Φ_{p1}-proof theory:

Proposition 3. *Let $\langle A, R\rangle$ be an argumentation framework, and let $S \subseteq A$. If E is a Φ_{p1}-proof for S, then $\cup_{d\in E}(\mathrm{PRO}(d))$ is a p-admissible set containing S. If $S \neq \emptyset$ is in a preferred p-extension of $\langle A, R\rangle$, and if A is finite, then there exists a Φ_{p1}-proof for S.*

Example 7 (Example 6 (contd)). $d_4 = [\mathrm{PRO}, a].[\mathrm{OPP}, b].[\mathrm{PRO}, c]$ is the unique ϕ_{p1}-proof for a. $d_5 = [\mathrm{PRO}, n].[\mathrm{OPP}, e].[\mathrm{PRO}, i]$ is the unique ϕ_{p1}-proof for n. $d_6 = [\mathrm{PRO}, i]$ is the unique ϕ_{p1}-proof for i.

Consider the set $S_1 = \{a, n\}$. $\{d_4, d_5\}$ is a set of ϕ_{p1}-proofs, one for each argument of S_1, but $\mathrm{PRO}(d_4) \cup \mathrm{PRO}(d_5) = \{a, c, n\}$ contains an indirect conflict between c and n. So $\{d_4, d_5\}$ is not a Φ_{p1}-proof for S_1, and since there is no other candidate set of ϕ_{p1}-dialogues, we prove again that S_1 is not credulously accepted.

Now consider the set $S_2 = \{a, i\}$. $\mathrm{PRO}(d_4) \cup \mathrm{PRO}(d_6) = \{a, c, i\}$ is without indirect conflict. So $\{d_4, d_6\}$ is a Φ_{p1}-proof for S_2.

A Φ_1-proof is a set of sequential proofs that shows not only why the set of arguments is acceptable, but also why each argument of the set is. However, it still does not show the lines of defence for each argument inside each proof.

To solve this drawback, we define a new legal-move function and proof theory.

4.2 Detailed Proof Theory

[10] proposed a legal-move function ϕ_2 such that each advanced argument attacks the immediately preceding one. Combined with the second winning criterion (winning strategy), it allows to design proof theories that make clear, for each argument of the proof, the lines of defence. We adapt this proof theory to our acceptance problem under the prudent preferred semantics, in order to define a proof theory for a set of arguments S that shows precisely the lines of defence for each argument of the proof. This proof theory takes the form of a coherent set of proofs for the acceptability of each argument of S.

Definition 13. *Given an argumentation framework $\langle A, R\rangle$, let $\phi_{p2} : A^* \rightarrow 2^A$ be defined by:*

- *if d is a dialogue of odd length (next move is by* OPP*),*

$$\phi_{p2}(d) = R^-(\mathrm{arg}(\mathrm{last}(d))) \setminus R^+(\mathrm{PRO}(d));$$

- *if d is a dialogue of even length (next move is by* PRO*),*

$$\phi_{p2}(d) = \phi_{p1}(d) = R^-(\mathrm{arg}(\mathrm{last}(d))) \cap \mathrm{POSSI}(d).$$

We define a notion of a ϕ_{p2}-proof for an argument only. This notion could be extended to a set of arguments, but since our goal here is to define an explicit proof theory that shows precisely the line of defense for each argument, we do not need to consider this extension.

Definition 14. *Let x be an argument that does not belong to* Refl. *A ϕ_{p2}-proof for x is a ϕ_{p2}-winning strategy D for $\{x\}$ such that* $\mathrm{PRO}(D)$ *is without indirect conflict.*

The ϕ_{p1}-proof theory for an argument is equivalent to the ϕ_{p2}-proof theory as shown by the following proposition. Thus, the latter too is sound and complete for the credulous acceptance problem [3].

Proposition 4. *There exists a ϕ_{p1}-proof for the argument x if and only if there exists a ϕ_{p2}-proof for x.*

Now we define a proof theory for a set of arguments:

Definition 15. *Let (A, R, ϕ_{p2}) be a dialogue type, and let $S \subseteq A$. A Φ_{p2}-proof for S is a set E of ϕ_{p2}-proofs such that the following two conditions hold:*

1. *for every $a \in S$, there exists one and only one $D \in E$ such that D is a ϕ_{p2}-proof for a, and*
2. $\cup_{D \in E}(\mathrm{PRO}(D))$ *is without indirect conflict.*

The following result establishes the soundness and the completeness of the Φ_{p2}-proof theory:

Proposition 5. *Let $\langle A, R\rangle$ be an argumentation framework, and let $S \subseteq A$. If E is a Φ_{p2}-proof for S, then $\cup_{D \in E}(\mathrm{PRO}(D))$ is a p-admissible set containing S. If $S \neq \emptyset$ is in a preferred p-extension of $\langle A, R\rangle$, and if A is finite, then there exists a Φ_{p2}-proof for S.*

Example 8. Let $AF_3 = \langle A, R\rangle$ as depicted on Figure 3. A ϕ_{p2}-proof for a should contain at least two ϕ_{p2}-dialogues: one that shows how a is defended against b, and another that shows how it is defended against c. $\delta_1 = [\mathrm{PRO}, a].[\mathrm{OPP}, c].[\mathrm{PRO}, n]$ is a ϕ_{p2}-dialogue about a won by PRO that shows how a is defended against c. $\delta_2 = [\mathrm{PRO}, a].[\mathrm{OPP}, b]$ is a ϕ_{p2}-dialogue about a that cannot be continued by PRO since the only attacker of b is i, and i indirectly attacks a. So δ_2 is not won by PRO, and since there exists no ϕ_{p2}-dialogue that shows that a can be defended against b, there exists no ϕ_{p2}-proof for a, and hence no Φ_{p2}-proof for $\{a\}$; a is not credulously accepted under the prudent preferred semantics.

Consider now an extension of AF_3 with an argument d such that d attacks b. There exists in this new framework a ϕ_{p2}-dialogue that shows that a can be defended against b: $\delta_3 = [\mathrm{PRO}, a].[\mathrm{OPP}, b].[\mathrm{PRO}, d]$, won by PRO. $D_1 = \{\delta_1, \delta_3\}$ is a ϕ_{p2}-winning strategy for a. Since $\mathrm{PRO}(D_1) = \mathrm{PRO}(\delta_1) \cup \mathrm{PRO}(\delta_3) = \{a, n, d\}$ is without indirect conflict, D_1 is a ϕ_{p2}-proof for a. Hence $\{D_1\}$ is a Φ_{p2}-proof for $\{a\}$, a is credulously accepted under the prudent preferred semantics.

Consider the set $S = \{a, n\}$ in the new framework. $\delta_4 = [\mathrm{PRO}, n]$ is a ϕ_{p2}-dialogue about n won by PRO. $D_2 = \{\delta_4\}$ is a ϕ_{p2}-proof for n. Since $\mathrm{PRO}(D_1) \cup \mathrm{PRO}(D_2)$ is without indirect conflict, $\{D_1, D_2\}$ is a Φ_{p2}-proof for S.

[3] This result is an extension of [10]'s Proposition 3.12 to the prudent preferred semantics.

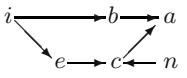


Fig. 3. Argumentation framework AF_3

4.3 Towards a Generalization to Other Acceptance Problems

The ϕ_{p1}-proof theory is adapted from the proof theory proposed in [11] for Dung's preferred semantics. As regards the Φ_{p1} and Φ_{p2}-proof theories, they could in turn be easily modified to answer the acceptance problem of a set of arguments under the preferred semantics.

More generally, the dialectical proof theories presented here rely upon two legal-move functions: ϕ_{p1} and ϕ_{p2}. These functions are adapted from [10]'s legal-move functions ϕ_1 and ϕ_2. The main difference between the two approaches lies in the definition of $\mathrm{POSSI}(d)$, that is, in the set of arguments PRO is allowed to play for pursuing a dialogue d. Actually, in [10], the goal was for PRO to build an admissible set, whereas PRO's goal here is to build a p-admissible set; the arguments to be excluded are the ones that would generate an indirect conflict, whereas for [10] it is those that would generate a direct conflict.

This small difference suggests that [10]'s legal-move functions and dialectical framework could be, as easily as for the prudent preferred semantics, adapted to any other semantics that is based on a variant of Dung's notion of admissibility. [16] has recently proposed a generalization of these variants in a *Constrained argumentation framework*. We plan to study how the dialectical framework and the proof theories presented here could be generalized to this framework.

5 Conclusion

Our contribution in this paper is twofold: first, we have proposed a dialectical framework that allows to design proofs for acceptance problems for a set of arguments. Second, we have proposed proof theories in this framework for the credulous acceptance problem for a set of arguments under the prudent preferred semantics.

Depending on the application where the acceptance problem would have to be solved, the one or the other of these theories would be used. For example, if a quick answer to the problem is needed, with a minimal proof, then the ϕ_{p1}-proof theory should be chosen. On the contrary, if a detailed, explicit proof of the answer is needed, no matter how long it is, then the Φ_{p2}-proof theory would be ideal.

Due to a lack of space, we have presented here neither the detailed proofs of our results, nor the algorithms that we have designed to build the dialectical proof theories proposed. These algorithms, and the proofs of our results, can be found in [17].

As a future work, we plan to study the generalization of this approach for credulous acceptance problems to Constrained argumentation frameworks. We also plan tostudy the problem of sceptical acceptance under the prudent preferred semantics, that is, determining whether a set of arguments is included in every preferred p-extension.

Acknowledgements

The authors acknowledge the reviewers for their helpful comments.

References

1. Prakken, A., Vreeswijk, G.: Logics for defeasible argumentation, vol. 4, pp. 219–318. Kluwer Academic Publishers, Dordrecht (2002)
2. Bondarenko, A., Dung, P.M., Kowalski, R., Toni, F.: An abstract, argumentation-theoretic approach to default reasoning. Artif. Intell. 93, 63–101 (1997)
3. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. 77(2), 321–358 (1995)
4. Bench-Capon, T.J.M.: Persuasion in practical argument using value-based argumentation frameworks. J. Log. Comput. 13(3), 429–448 (2003)
5. Baroni, P., Giacomin, M., Guida, G.: Extending abstract argumentation systems theory. Artif. Intell. 120(2), 251–270 (2000)
6. Coste-Marquis, S., Devred, C., Marquis, P.: Prudent semantics for argumentation frameworks. In: ICTAI'05, pp. 568–572 (2005)
7. Coste-Marquis, S., Devred, C., Marquis, P.: Inference from controversial arguments. In: Sutcliffe, G., Voronkov, A. (eds.) LPAR 2005. LNCS (LNAI), vol. 3835, pp. 606–620. Springer, Heidelberg (2005)
8. Cayrol, C., Lagasquie-Schiex, M.C.: On the acceptability of arguments in bipolar argumentation frameworks. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 378–389. Springer, Heidelberg (2005)
9. Vreeswijk, G., Prakken, H.: Credulous and sceptical argument games for preferred semantics. In: JELIA'00, pp. 224–238 (2000)
10. Cayrol, C., Doutre, S., Mengin, J.: On decision problems related to the preferred semantics for argumentation frameworks. J. Log. Comput. 13(3), 377–403 (2003)
11. Doutre, S., Mengin, J.: On sceptical versus credulous acceptance for abstract argument systems. In: Alferes, J.J., Leite, J.A. (eds.) JELIA 2004. LNCS (LNAI), vol. 3229, pp. 462–473. Springer, Heidelberg (2004)
12. Dunne, P.E., Bench-Capon, T.J.M.: Two party immediate response disputes: Properties and efficiency. Artif. Intell. 149(2), 221–250 (2003)
13. Amgoud, L., Cayrol, C.: A reasoning model based on the production of acceptable arguments. Annals of Math. and Artif. Intell. 34(1-3), 197–215 (2002)
14. Prakken, H.: Coherence and flexibility in dialogue games for argumentation. J. Log. Comput. 15(6), 1009–1040 (2005)
15. Amgoud, L., Maudet, N., Parsons, S.: Modelling dialogues using argumentation. In: ICMAS'00, pp. 31–38 (2000)
16. Coste-Marquis, S., Devred, C., Marquis, P.: Constrained argumentation frameworks. In: KR'06, pp. 112–122 (2006)
17. Devred, C.: Proof theories and algorithms for the credulous derivation problem under the prudent preferred semantics. Technical report, CRIL - Université d'Artois (2007)

Towards an Extensible Argumentation System

Adam Z. Wyner and Trevor J.M. Bench-Capon

Department of Computer Science
University of Liverpool
Ashton Building, Ashton Street
Liverpool, L69 3BX, United Kingdom
{azwyner, tbc}@csc.liv.ac.uk
http://www.csc.liv.ac.uk

Abstract. Many types of inter-agent dialogue, including information seeking, negotiation and deliberation can be seen as varieties of argumentation. Argumentation is especially appropriate where demonstration is not possible because the information is incomplete and uncertain or because the parties involved in the argument have different perspectives on an issue. Argumentation frameworks provide a powerful tool for evaluating the sets of conflicting arguments which emerge from such dialogues. Originally argumentation frameworks considered arguments as completely abstract entities related by a single attack relation, which always succeeded. Use of the frameworks in practical applications such as law, e-democracy and medicine has motivated a distinction between successful and unsuccessful attacks, determined by properties of the conflicting arguments. This remains insufficient to capture a range of phenomena which arise from procedural and contextual considerations. These require that a successful attack depend not only on the properties of the conflicting arguments but also on the nature of the attack and the context in which it is made. In this paper we present an analysis of arguments, their properties and relations which can accommodate a wide range of such phenomena. Our analysis is extensible for we can add components to each system while preserving an overarching argumentation framework. We first capture the abstract notions of original argumentation frameworks, and then introduce a system which embraces properties of arguments. This system is further extended in two ways to include properties of relations between arguments. We illustrate each system with a characteristic example and discuss the particular features of argumentation which they can address.

Keyword: argumentation.

1 Introduction

Many types of inter-agent dialogue, including information seeking, negotiation and deliberation can be seen as varieties of argumentation. Argumentation is especially appropriate where demonstration is not possible, through incompleteness and uncertainty of information, or because the parties in the dialogue differ

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 283–294, 2007.

in their beliefs, perspectives or interests. Argumentation frameworks (AF), introduced by Dung [1], provide a tool for evaluating the sets of conflicting arguments which emerge from such dialogues. In [1], arguments are entirely abstract, and they are related by a single binary relation, the *attacks* relation between arguments (for an alternative, see [2]); arguments and attacks are homogenous, and attacks always succeed. The status of arguments is evaluated relative to a subset of the arguments in the framework. An argument is acceptable with respect to such a set if it is not attacked by any member of that set, and any arguments which attack it are attacked by some member of the set. A considerable amount of useful theoretical work has addressed issues arising from Dung's AF such as the different semantics that can be described when differently characterised subsets of the AF are used to defend arguments as well as the complexity and algorithmic issues relating to decision problems within these AFs [3].

The level of abstraction used by Dung is appropriate for some purposes, such as the arguments of logic and mathematics. However, the framework is too abstract in contexts where it is acknowledged that arguments are not homogenous: for example, we speak of weak and strong arguments, where it is helpful to distinguish between an argument attacking another argument, and an argument being sufficiently strong relative to the argument being attacked for that attack to successfully defeat that argument. Accordingly, there have been proposals which ascribe properties to arguments which can be used to represent these relative strengths (e.g. preference based AFs (PAFs) of [4] and value based AFs (VAFs) of [5]). In PAFs and VAFs, the properties are used to *filter* attacks: effectively, those attacks which are regarded as unsuccessful are removed. Once the properties have played their part, the framework can then be regarded as a Dung-style AF, and thus these approaches can benefit from the theoretical results applying to AFs.

PAFs and VAFs have extended the range of problems to which AFs are appropriate. However, they cannot handle other situations. For example, both PAFs and VAFs use a single ordering on preferences or values with respect to which the attacks are filtered. Yet in some cases, the various interested parties may not agree on an order for preferences or values. Instead, who is empowered to resolve a given conflict may depend on the types of the arguments involved. An example is the English legal system in which juries decide matters of fact and judges decide matters of law. These examples relate to properties of *arguments*, but it may also be that *attacks* are not homogenous and need to be differentiated. Some types of attack may refute an argument while others may only cast some doubt, which can, in the appropriate circumstances, be disregarded. For example, we may decide that while rebutting attacks must always be respected, undercutting attacks can be ignored under some kind of *ceteris paribus* assumption.

These are but two examples of situations which we want analyze in terms compatible with an abstract AF by adding subsystems which are *refinements* of an AF and enable information necessary to the resolution of the dispute to be represented and considered. As the subsystems can be *extended* with additional predicates, relations, and functions, we call it an *extensible argumentation*

system. The additional information only filters the attack relation, thus reducing the subsystem to an abstract AF, which we do not modify. As these subsystems are *refinements* of an abstract AF, we maintain our understanding of them at that abstract level. Moreover, as the subsystems can be related to one another, we have a unified approach to meeting the disparate needs for additional information, and so avoid the production of new systems in an ad hoc manner each time a new phenomenon is encountered.

The paper is structured as follows. First we describe $\mathbf{ExArS}_0$, an **Ex**tensible **Ar**gumentation **S**ystem, which is simply Dung's original AF described in our typed, functional decomposition style. This provides both the foundation for further extension and provides an introduction to our notation. We introduce this style of notation because it provides a clear computational model, facilitates specification of complex functions out of basic components and functions, and is straightforward to implement in a functional programming language. It can also be used to prove properties of the model, although that is not the focus of this paper. Remaining with a familiar extension, we then describe $\mathbf{ExArS}_1$, which corresponds to a VAF. A very similar extension would lead to PAF. The next extension, $\mathbf{ExArS}_2$, is novel, and provides the machinery needed to handle the situation where different parties to a deliberation are given responsibility to decide conflicts on arguments of different types. Finally, we introduce a second novel extension, $\mathbf{ExArS}_3$, which distinguishes different kinds of attack. In $\mathbf{ExArS}_3$ we restrict ourselves to the well known varieties of attack (rebuttal, undercut and premise defeat), but further extension could be made to deal with the increased range of attacks found in, for example, the Carneades system [6]. We conclude the paper with some discussion of areas of argumentation which could motivate further extensions.

2 An Extensible Argumentation System

An **Ex**tensible **Ar**gumentation **S**ystem (**ExArS**) is comprised of argument objects, relations, and definitions of auxiliary concepts. Here, we provide only those components of the system as are needed to make our point before extending it to account for some other phenomena. We use subscripting to differentiate the systems. Our assumption is that [1] is the most abstract system. For clarity, all elements are subscripted to indicate the extensible system to which they belong.

We assume `boolean` and `object` types. The `boolean` type has two subsorts: `true` and `false`. Initially, we have but one sort of *object* type, namely, arguments. Expressions of the form $\texttt{object} \rightarrow \texttt{boolean}$ are to be understood as functions from objects to truth-values; that is, in this instance, it expresses the *characteristic function* such that an expression of that type denotes a *set of objects*. By the same token, $(\texttt{object} \times \texttt{object}) \rightarrow \texttt{boolean}$ denotes a set of ordered pairs of objects, which is a relation. The most basic system only has a set of arguments, the *nodes*, and a single *attack* relation between arguments, the *arcs* between the nodes.

Definition 1. ***ExArS***$_0$

- *Arg*$_0$*, a set of argument names,* a_1*,...,*a_n*, of type* `argument` *which denote arguments.*
- *ArgAtt*$_0$*, an attack relation defined on Arg*$_0$*, where we read ArgAtt*$_0$*(*a_1*,* a_2*) as argument* a_1 *attacks*$_0$ *argument* a_2*. ArgAtt*$_0$ *is of type (*`argument` $\times$ `argument`*)* $\rightarrow$ `boolean`*. We assume that no argument attacks itself:* $\forall x \in Arg_0$ $\neg ArgAtt_0(x, x)$*.*

The assumption that arguments do not attack themselves follows [3]; it is optional and not crucial to our discussion.

We give a sample of the main auxiliary definitions of [1], which are notational variants of the original definitions. Suppose S is a subset of Arg_0.

Definition 2. ***Acceptability, Admissibility, and Extensions***

- $x \in Arg_0$ *is acceptable*$_0$ *to S if:* $\forall y \in Arg_0$ *where* $ArgAtt_0(y, x)$*,* $\exists z \in S$ *where* $ArgAtt_0(z, y)$*.*
- *S is conflict-free*$_0$ *if:* $\forall x, y \in S$ $\neg ArgAtt_0(x,y)$*.*
- *A conflict-free*$_0$ *set S is admissible*$_0$ *if:* $\forall x \in S$*, x is acceptable*$_0$ *to S.*
- *S is a preferred extension*$_0$ *if: S is a maximal admissible*$_0$ *set.*

Definitions for *stable extension*, *coherence*, *credulously accepted*, and *skeptically accepted* follow suit. Note that context distinguishes between the *mathematical* sense of *extension* as in *preferred extension* from the sense as in *enlarging the scope or operation* of an AF.

3 First Extension – Value-Based Argumentation

The Value-based argumentation framework of [5] builds on [1]. The principal intuition is that an argument attack may succeed or fail *relative to a value that is ascribed to an argument*; that is, intuitively, if there is an argument that I should go eat pastry, and it is attacked by an argument that I should diet, I might still accept that I should go eat pastry because I value pleasure more than fitness. Here we provide it as our first example of an extension to $\mathbf{ExArS}_0$; it is a notational variant of [5]. We only provide some of the key clauses.

Definition 3. ***ExArS***$_1$

- *Arg*$_1$*, a set of arguments.*
- *ArgAtt*$_1$*, an attack relation defined on Arg*$_1$*.*
- *Val is set of value names,* v_1*,...,*v_n*, of type* `value`*, which denote values. We assume Val*$_1$ *is defined for* ***ExArS***$_1$*.*
- *RankingScheme is a total ordering of Val; it is is understood as an audience. For x,* $y \in$ *Val, if RankingScheme(x, y), we say that x is preferred to y in the audience RankingScheme. A RankingScheme is of type (*`value` $\times$ `value`*)* $\rightarrow$ `boolean`*. We assume a RankingScheme*$_1$ *is defined for* ***ExArS***$_1$*.*

In addition to these components, arguments are assigned a value.

Definition 4. ***Argument-Value Ascription***

$\forall x \in Arg_1 \; \exists y \in Val_1$ $AssignArgVal(x) = y$, where $AssignArgVal$ is a function from arguments to values. We assume $AssignArgVal_1$ is defined for $\boldsymbol{ExArS}_1$.

Definitions of argument defeat, acceptability, conflict-free, and admissibility are relativized to audiences, which rank the values of the arguments. Notice, in particular, that the success or failure of an argument attack is *determined with respect to a property ascribed to arguments*. Suppose S is a subset of Arg_1.

Definition 5. ***Defeat, Acceptability, and Admissiblity***

- *For arguments $x, y \in Arg_1$, x $defeats_1$ y with respect to $RankingScheme_1$ if: $ArgAtt_1(x,y) \wedge \neg RankingScheme_1(AssignArgVal_1(y), AssignArgVal_1(x))$*
- *$x \in Arg_1$ is $acceptable_1$ to S if: $\forall y \in Arg_1$ that $defeats_1$ x, $\exists z \in S$ that $defeats_1$ y.*
- *S is $conflict\text{-}free_1$ if: $\forall x, y \in S$ [$\neg ArgAtt_1(x,y)$ or $\neg RankingScheme_1(AssignArgVal_1(x), AssignArgVal_1(y))$]*
- *A $conflict\text{-}free_1$ set S is $admissible_1$ if: $\forall x \in S$, x is $acceptable_1$ to S.*

As with the Dungian framework, we can define notational variants of the notions of *preferred extensions*, *sceptically* and *credulously acceptable*, as well as notions relating to value orders such as *objectively* and *subjectively* acceptable as in [5].

To this point, we have but recast familiar argumentation frameworks into our language. The main advantage, as shown in the subsequent section, is that we can then *extend* the basic components of these frameworks to address a range of additional issues and problems in the argumentation literaure while keeping the basics of the framework intact. For example, in a VAF, the additional information about values is used to *filter* the attack relation, leaving only *successful* attacks (i.e. those in which an argument is *defeated*) to be used in the calculation of *admissible sets*. Thus, the extended system reduces to an abstract AF. Furthermore, we have done so in a manner consistent with an abstract AF, so the relationship between the abstract AF and the extension is clear and not ad hoc. Our framework also provides a means to compare and contrast argumentation proposals.

4 Second Extension – Adjudication

In this section, we extend $\mathbf{ExArS}_1$ to account for issues in argumentation which have not previously been accounted for in a Dungian style analysis. Note that it can, but need not, be that the extensions add to or further specify previous extensions; the key point for our purposes is just that *every extension extends the abstract AF of* $\boldsymbol{ExArS}_0$ *in such a way as to add filters on the attack relation.* In this extension, we have *multiple audiences*, which may themselves be ordered. We also *differentiate* attack relations, which is to label the arcs. Using the attack relation in this way is novel in the AF literature.

In $\mathbf{ExArS}_1$, we had but one ranking of values (i.e. one audience), and attacks succeeded or failed with respect to that ranking and the values of arguments. However, in an argument, there may be two or more audiences, which means there are two (or more) different rankings of the values [7]. In this case, we must consider *Multi-agent Systems*, where we associate audiences with agents. For instance, we can label one audience for a government official *GovtOff* and another for a religious minister *RelMin*: for values v_i and v_j, in $\mathrm{RankingScheme}_{GovtOff}$, v_i is preferred to v_j while in $\mathrm{RankingScheme}_{RelMin}$, v_j is preferred to v_i. Clearly, these conflict. Any two (or more) distinct audiences could be so represented. Moreover, the different audiences may have different capacities to argue about the outcome of an attack. For example, suppose that the values are v_i = *economic well-being* and v_j = *spiritual well-being*; with respect to some particular issues, the government official's valuation of the arguments is paramount, while in others it is the religious minister's, leaving aside how such determinations are made. We call this *adjudication*, for a means to resolve a conflict between competing value systems is provided.

To account for such cases, we subsort the ranking schemes and the attack relations, where one subsort represents the *GovtOff* and another the *RelMin*. The outcome of the attack relation is relative to the label on the arc and the associated ranking scheme: if the arc is labelled with *GovtOff*, then we use the ranking scheme for the *GovtOff* in order to determine the outcome of the attack; if the arc is labelled with *RelMin*, then we instead use the ranking scheme for the *RelMin*.

For clarity, we provide the extension along with an example of two ranking schemes. We turn to the attack relations in a moment.

Example 1. $\mathbf{ExArS}_2$

- Arg_2 is $\{\mathrm{a}_1, \mathrm{a}_2, \mathrm{a}_3, \mathrm{a}_4\}$.
- ArgAtt_2 is $\{<\mathrm{a}_1, \mathrm{a}_2>, <\mathrm{a}_2, \mathrm{a}_1>, <\mathrm{a}_2, \mathrm{a}_3>, <\mathrm{a}_3, \mathrm{a}_4>, <\mathrm{a}_4, \mathrm{a}_3>\}$.
- Val_2 is $\{\mathrm{v}_1, \mathrm{v}_2\}$.
- $\mathrm{RankingSchemeSet}_2$ is a set of elements which are of type `RankingScheme` and total orderings of Val_2.
- $\mathrm{AssignArgVal}_2$ is a function from elements of Arg_2 to elements of Val_2.
- $\mathrm{RankingScheme}_{RelMin} \in \mathrm{RankingSchemeSet}_2$ is $\{<\mathrm{v}_2,\mathrm{v}_1>\}$
  $\mathrm{RankingScheme}_{GovtOff} \in \mathrm{RankingSchemeSet}_2$ is $\{<\mathrm{v}_1,\mathrm{v}_2>\}$.

We have variables of type `RankingScheme`. Suppose the values of arguments are:

Example 2. **Argument-Value Ascription**

- $\mathrm{AssignArgVal}_2(\mathrm{a}_1) = \mathrm{v}_1$
- $\mathrm{AssignArgVal}_2(\mathrm{a}_2) = \mathrm{v}_2$
- $\mathrm{AssignArgVal}_2(\mathrm{a}_3) = \mathrm{v}_1$
- $\mathrm{AssignArgVal}_2(\mathrm{a}_4) = \mathrm{v}_2$

The key novelty in this extension is the introduction of *labels* for the arcs, which subsort the attack relation.

Example 3. **Subsorts of Argument Attacks**

- ArgAttSet_2 is a set of subsets of ArgAtt_2.
- $\text{ArgAtt}_{RelMin} \in \text{ArgAttSet}_2$ is $\{<a_1,a_2>, <a_2,a_1>\}$
- $\text{ArgAttacks}_{GovOff} \in \text{ArgAttSet}_2$ is $\{<a_2,a_3>, <a_3,a_4>, <a_4,a_3>\}$

The subsorts of attacks represent who has the *control* of the attack. Depending on which agent has control of the attack, we relativize the ranking scheme to that agent's values. Thus, we express *which agent's values determine the success or failure of the attack*. We assume a function from arcs to ranking schemes:

Definition 6. ***Function from Arcs to Ranking Schemes***

ArcRankFun is a function from arcs to ranking schemes of type $<$`argument`, `argument`$>$ $\rightarrow$ `RankingScheme`. *Given arguments x,* $y \in Arg_2$, $ArcRankFun(x, y) \in RankingScheme_z$, *where* $RankingScheme_z \in RankingSchemeSet_2$, *and* $ArgAtt_z(x, y)$, *where* $ArgAtt_z \in ArgAttSet_2$. *The subscript z associates arc controllers with ranking schemes. Thus, (ArcRankFun (x, y)) (*$AssignArgVal_2$ *(x),* $AssignArgVal_2$ *(y)) is* ***true*** *or* ***false*** *relative to a ranking scheme and controller of an arc. We assume an* $ArcRankFun_2$ *is defined for* $\boldsymbol{ExArS_2}$.

We have given an example of how attack relations are relativized, but additional definitions may be required such as *partitioning* the attack relations to avoid conflicts between the agents or ordering the attack relations, giving priority to one over the other. Such additions could be used to define *procedural contexts*, which will be left to future work.

With this, our definitions for notions such as argument defeat and admissibility are relativized to the ordered values of the audience, supposing S is subset of Arg_2.

Definition 7. ***Defeat, Acceptability, and Admissiblity***

- *For arguments* $x, y \in Arg_2$, x $defeats_2$ y *with respect to the values of the agent which controls that arc if:* $ArgAtt_2(x,y) \wedge \neg(ArcRankFun_2\ (x, y))\ (AssignArgVal_2(y),\ AssignArgVal_2(x))$.
- $x \in Arg_2$ *is* $acceptable_2$ *to S if:* $\forall y \in Arg_2$ *that* $defeats_2$ x, $\exists z \in S$ *that* $defeats_2$ y.
- S *is* $conflict\text{-}free_2$ *if:* $\forall x, y \in S\ [\ \neg ArgAtt_2(x,y)$ *or* $\neg\ (ArcRankFun\ (x, y))\ (AssignArgVal_2(x),\ AssignArgVal_2(y))]$.
- *A* $conflict\text{-}free_2$ *set S is* $admissible_2$ *if:* $\forall x \in S$, x *is* $acceptable_2$ *to S.*

Given these definitions and examples 1-3, the set $\{a_2, a_3\}$ is admissible_2 in $\mathbf{ExArS}_2$. If we had used only one of the rankings, as we would have been obliged to do in $\mathbf{ExArS}_1$, either would have given a different result. However, we would not be able to represent distinct controls over attacks. Using the specification, we can filter the attack relations to get back to an abstract AF structure; in doing so, we can homogenize the extension with the abstract Dungian analysis.

5 Third Extension – Internal Structure of Arguments

In some argumentation theories ([8], [6]), argument objects are related to statements and have a *mereological* (i.e. part) structure, where arguments have statements which are assumptions, a statement which is a conclusion, and a reasoning relation between the assumptions and conclusion.[1]. With such structure, we can represent fine-grained argument attacks such as attacks on assumptions, conclusions, and reasoning relations as found in common-sense argumentation. This is relevant not only to incorporate well-known approaches to argumentation into a Dungian style framework (e.g. Toulmin Structures), but more importantly to provide an analysis of *procedural contexts*, wherein different sorts of arguments and argument attacks are allowable in a given context. Some approaches to argumentation which make use of structured arguments (e.g. [6]) cannot be characterized as an extension of $\mathbf{ExArS}_0$ as they do not provide definitions for attack or admissible sets of arguments. As we do provide such definitions, we can *homogenize* such arguments to a Dungian analysis, which is a novel analysis.

For our purposes, an *argument* has *assumptions*, a *conclusion*, and a *reasoning relation*. Arguments are in relation to statements. The reasoning relation is the conditional as used in Defeasible Logic [9]. A variety of *attack* relations are defined with respect to the *part* of the argument under attack. These attacks correspond to familiar notions of *rebuttal*, *undercutting* and *assumption defeat*.

We do not need values or multiple agents as in $\mathbf{ExArS}_2$; the extensions do not strictly need to extend any extension other than $\mathbf{ExArS}_0$, though they can. We introduce statements and reasoning relations as first-class objects which are in relation to an argument.

Definition 8. ***ExArS***$_3$

- *Arg_3 is a set of arguments $\{a_1, \ldots, a_n\}$.*
- *$Stat_3$ is a set of atomic statement names $\{s_1, \ldots, s_n\}$, which denote atomic propositions and are of type* `statement`*. If s is a statement, then $\neg s$ is a statement. In no model can s and $\neg s$ both hold in any context; s and $\neg s$ are called contraries. $\neg s$ is the only complex statement.*
- *$ArgAtt_3$, an attack relation defined on Arg_3.*
- *$ReasRel_3$ is a set of reasoning relation names $\{r_1, \ldots, r_n\}$, which denote reasons and are of type* `reasoning relation`*.*

The assumptions and reasoning relations are related to the argument.

Definition 9. ***Argument Relations***

- *If $s \in Stat_3$ and $a \in Arg_3$, then Assum(a,s) is a well-formed relation. It is read as the statement s is an assumption of argument a. The assumption relation is of type (*`argument` $\rightarrow$ `statement`*) $\rightarrow$* `boolean`*; it is a many-to-many relation.*

[1] In other approaches [6], there is a different typology, which is not relevant in this paper.

– *If $a \in Arg_3$ and $r \in ReasRel_3$, then ReasRelFunc(a) = r is a function from arguments to reasoning relations. It is read as the reasoning relation of argument a is r. The function is of type* `argument` → `reasoning relation`.

To specify a conclusion, we first define the set of assumptions.

Definition 10. ***Set of Assumptions***

λs Assum(a,s) is the set of statements which are assumptions of a given argument a, where $a \in Arg_3$ and $\forall\ s \in \lambda s$ Assum(a,s), $s \in Stat_3$.
It is a set of type `statement` → `boolean`.

A conclusion is a statement which is functionally related to the argument, assumptions, and reasoning relation. For our purposes here, we assume an argument only has *one* conclusion just as it has only *one* reasoning relation.

Definition 11. ***Conclusion Function***

For $a \in Arg_3$, λs Assum(a,s), ReasRelFunc(a), and $s \in Stat_3$,
Conclusion(a) = s is a function from an argument to an implied statement s given the argument's assumptions and reasoning relation.

For brevity, we make the following assumptions without formally specifying them. Two arguments a_1 and a_2 are *identical* when they have the same assumptions, conclusions, and reasoning relations. Furthermore, an argument a_1 is a *subargument* of another argument a_2 if the conclusions and reasoning relations of a_1 are the same as a_2, but the set of assumptions of a_1 is a proper subset of a_2. Finally, given two arguments with the same assumptions and reasoning relation, the same conclusion must follow. With them, we have the following:

Definition 12. ***Argument Distinction***

$\forall x,\ y \in Arg_3$ DistinctArg(x,y) if: x and y are not identical arguments and neither is a subargument of the other.

In contrast to previous Dungian analyses, we can *analytically* define the notion of attack: the arguments are not only distinct, but their conclusions are contraries.

Definition 13. ***General Argument Attack***

Where $ArgAtt_3 \subseteq (Arg_3 \times Arg_3)$, $\forall x,\ y \in Arg_3$ $ArgAtt_3$(x,y) if:
DistinctArg(x,y) $\wedge$ Conclusion(x) = $\neg$Conclusion(y).

This definition correlates to the more familiar *rebuttal* attack; it claims that *any* attack of one argument on another is at least an *attempt* to rebut. In contrast, it is unclear in virtue of what one argument attacks another in [1], and by the same token, in virtue of what arguments hold together in an admissible set.

We have defined the most general sort of attack. However, we can have *subsorts of attacks keyed to the mereological structure of the arguments*, which is

an analytic basis for labelling the attack arcs. For our purposes here, we can have attacks on assumptions or attacks on rules, though one could define other sorts of attacks given other subproperties of arguments. In effect, the subsort of attack expresses *why the conclusion is denied*.

An *attack on the reasoning relation* means that where one argument attacks another, the arguments *differ in terms of the reasoning relation*. This correlates to the more familiar *undercutting* attack.

Definition 14. ***Reason Relation Attack***

Where $ReasonRelAtt \subset ArgAtt_3$, $\forall\ x,\ y \in Arg_3$ *ReasonRelAtt(x, y) if:* $ReasonRelFun(x) \neq ReasonRelFun(y)$.

This sort of attack specifies that we do not accept the conclusion of the attacked argument because we do not accept the reasoning relation which led to the conclusion.

An *attack on an assumption* means that the assumption of one argument is the contrary of the *conclusion* of another argument. This correlates to the more familiar *premise defeat* (i.e. assumption defeat).

Definition 15. ***Assumption Attack***

Where $AssumAttack \subset ArgAtt_3$, $\forall y,z \in Arg_3$ $AssumAttack(y,z)$ *if:* $\exists x\ [x \in \lambda s(Assum(z,s)) \wedge Conclusion(y) = \neg x]$

To this point, we have defined subsorts of attacks in terms of reasoning relations and assumptions. Given additional properties ascribed to arguments, we could define further subsorts of attacks such as attacks on *presuppositions* or *exceptions* as in [6]; as well, we could *analytically* define *support* of one argument by another.

We can use such subsorts of attack to define different *procedural contexts*. For example, in reasoning about the economy, there may be a rule which can be undercut in certain contexts. However, in a particular model, one might abstract to *normal* contexts, and since there are no exceptions to the rules in normal contexts, undercutting can be ignored. Thus, we model procedural contexts where we understand them as contexts in which certain sorts of attacks can be applied, while other attacks are ruled out or ignored.

Given subsorts of attacks, we can relatively define *defeat, acceptability, conflict-free, and admissibility* in a variety of ways. For instance, with **General Argument Attack**, we have definitions similar to [1], supposing S is a subset of Arg_3.

Definition 16. ***Defeat, Acceptability, and Admissiblity***

- *For* $x,\ y \in Arg_3$, x *defeats*$_3$ y *with respect to contrary conclusions if:* $ArgAtt_3(x,y)$.
- $x \in Arg_3$ *is acceptable*$_3$ *to* S *if:* $\forall y \in Arg_3$ *that defeats*$_3$ x, $\exists z \in S$ *that defeats*$_3$ y.
- S *is conflict-free*$_3$ *if:* $\forall x,\ y \in S\ \neg ArgAtt_3(x,y)$.

– *A conflict-free$_3$ set S is admissible$_3$ if:* $\forall x \in S$, *x is acceptable$_3$ to S.*

Definitions for *preferred extension* and other semantic notions follow suit. Alternatively, instead of defining these notions based on `argAtt`, we could define a notion of defeat with respect to assumptions.

Definition 17. ***Defeat with respect to Assumptions***

For arguments x, $y \in Arg_3$, *x defeats$_3$ y with respect to asssumptions: AssumAttack(x,y)* $\wedge \neg\exists z \in Arg_3$ *AssumAttack(z,x).*

We can similarly define defeat with respect to rules. Another approach would be to impose an ordering on the attack relations, so making the defeat of an argument depend on an additional ordering parameter. Clearly, other notions can be defined given the different ways that arguments can attack and defeat one another. However, we leave further refinement and application for future research.

As with the previous extensions, this extension provides more fine-grained ways to define the attack relation and the defeats relation. However, once the defeat relation is determined for a particular set of arguments, we can abstract to the Dungian level of analysis. Thus, the examples in [6], for instance, can be accommodated in this framework.

6 Discussion

The paper is based on a range of sources ([1], [5], [6], and [10]). Our aim has been to adopt and adapt them into a cohesive and coherent formal argumentation system, while retaining the key observations and analyses of each. The key novel contributions of the paper are in two areas. First, we have provided a general format to extend the Dungian Framework in a number of fruitful directions to account for an additional spectrum of problems in argumentation. Second, we have introduced and applied *labelled arcs* in two extensions, showing how these can be used to represent and reason about complex issues in argumentation.

For future work, we mention how our system can be extended to *critical questions* [10]. First, we assume a notion of *supporting* arguments in AFs as have been introduced in [11]. To model critical questions, we subsort the attacks with respect to further subproperties of arguments, for instance, an *argument from expert opinion.* Where an argument from expert opinion is under discussion, we have the critical question *Is the expert really qualified to offer an opinion on the case at hand?* Following [12], where questions *denote* the set of statements which answer the question, we can suppose this question denotes *The expert is qualified...*, an argument which supports an assumption of the argument under discussion, and *The expert is not qualified...*, an argument which attacks it. Whether the argument is defeated depends on which of these arguments is sustained.

7 Conclusion

In this paper, we have shown how the Dungian Argumentation Framework can be extended in a variety of ways to address additional aspects of argumentation which had not previously been provided for. The manner of the extensions allows these additional aspects to be presented in a uniform and consistent way. Key among the extensions is the creation of *labelled arcs* which represent attack relations. With such arcs, we can distinguish sorts of attacks, which leads to a range of different ways to define admissible sets of arguments.

Acknowledgments

During the writing of this paper, the first author was supported by the Estrella Project (The European project for Standardized Transparent Representations in order to Extend Legal Accessibility (Estrella, IST-2004-027655)).

References

1. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77(2), 321–358 (1995)
2. Bochman, A.: Collective argumentatioin and disjunctive logic programming. Journal of Logic and Computation 13(3), 405–428 (2003)
3. Dunne, P.E., Bench-Capon, T.J.M.: Coherence in finite argument systems. Artif. Intell. 141(1), 187–203 (2002)
4. Amgoud, L., Cayrol, C.: On the acceptability of arguments in preference-based argumentation. In: Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98), pp. 1–7. Morgan Kaufmann, San Francisco (1998)
5. Bench-Capon, T.J.M.: Persuasion in practical argument using value-based argumentation frameworks. J. Log. Comput. 13(3), 429–448 (2003)
6. Gordon, T., Walton, D.: The carneades argumentation framework: Using presumptions and exceptions to model critical questions. In: Dunne, P.E., Bench-Capon, T. (eds.) Computational Models of Argument: Proceedings of COMMA 2006, pp. 195–207. IOS Press, Amsterdam (2006)
7. Bench-Capon, T., Doutre, S., Dunne, P.E.: Audiences in argumentation frameworks. Artificial Intelligence 171, 42–71 (2007)
8. Prakken, H.: Logical Tools for Modelling Legal Argument. In: A Study of Defeasible Reasoning in Law, Kluwer Academic Publishers, Dordrecht (1997)
9. Governatori, G., Maher, M.J., Antoniu, G., Billington, D.: Argumentation semantics for defeasible logic. Journal of Logic and Computation 14(5), 675–702 (2004)
10. Atkinson, K.: What Should We Do?: Computational Representation of Persuasive Argument in Practical Reasoning. PhD thesis, Department of Computer Science, University of Liverpool, Liverpool, United Kingdom (2005)
11. Cayrol, C., Devred, C., Lagasquie-Schiex, M.C.: Handling controversial arguments in bipolar argumentation systems. In: Dunne, P.E., Bench-Capon, T.J. (eds.) Computational Models of Argument, pp. 261–272. IOS Press, Amsterdam (2006)
12. Karttunen, L.: Syntax and semantics of questions. Linguistics and Philosophy 1(1), 3–44 (1977)

Dialectical Explanations in Defeasible Argumentation*

Alejandro J. García, Nicolás D. Rotstein, and Guillermo R. Simari

Artificial Intelligence Research and Development Laboratory
Department of Computer Science and Engineering
Universidad Nacional del Sur – Bahía Blanca, Argentina
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
{ajg,ndr,grs}@cs.uns.edu.ar

Abstract. This work addresses the problem of providing explanation capabilities to an argumentation system. Explanation in defeasible argumentation is an important, and yet undeveloped field in the area. Therefore, we move in this direction by defining a concrete argument system with explanation facilities.

We consider the structures that provide information on the warrant status of a literal. Our focus is put on argumentation systems based on a dialectical proof procedure, therefore we study *dialectical explanations*. Although arguments represent a form of explanation for a literal, we study the complete set of dialectical trees that justifies the warrant status of a literal, since this set has proved to be a useful tool to comprehend, analyze, develop, and debug argumentation systems.

1 Introduction

There has been attention focused on the role of explanations from several areas of Artificial Intelligence –especially from the expert systems community [1,2,3]. A few of them treat explanations in relation with argument systems [4]. In the literature, often an argument is regarded as an explanation for a certain literal. That is, the claim being explained is put under discussion, and only then it will be accepted or not. In belief revision, the role of explanations has also been studied [5]: a new perception is accompanied by an explanation, which is used (when needed) to resolve inconsistency with the agent's current beliefs. The piece of knowledge having the "best" explanation is the one that prevails, and is accepted as a new belief.

We are concerned with the type of explanations that give the necessary information to understand the warrant status of a literal. Since our focus is put on argumentation systems based on a dialectical proof procedure, we study *dialectical explanations* (from now on, δ-Explanations). Although we recognize arguments as an explanation for a literal, we are interested in obtaining the complete set of dialectical trees that justify the warrant status of a literal. We show how δ-Explanations can be a useful tool to comprehend and analyze the interactions among arguments, and for aiding in the encoding and debugging of the underlying knowledge base. Several examples, generated with an implemented system that returns, for a given query, both the answer and the associated δ-Explanation, are given throughout the paper.

* Partially supported by Universidad Nacional del Sur, CONICET (PIP 5050), and Agencia Nacional de Promoción Científica y Tecnológica.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 295–307, 2007.

An interesting review about explanations in heuristic expert systems is given in [1], in which a definition is given: "*...explaining* consists in *exposing something* in such a way that it is *understandable* for the receiver of the explanation –so that he/she improves his/her knowledge about the object of the explanation– and *satisfactory* in that it meets the receiver's expectations." In our approach, we *explain* through *exposing* the whole set of dialectical trees related to the queried literal. We believe that this information is *understandable* from the receiver's point-of-view, because all the arguments built, their statuses (*i.e.*, defeated/undefeated), and their interrelations are explicitly shown. This type of information would be *satisfactory* for the receiver, because it contains all the elements at stake in the dialectical analysis that supports the answer.

An empirical analysis about the impact of different types of explanations in the context of expert systems is given in [2]. The typology there described includes: 1) *trace:* a record of the inferential steps that led to the conclusion; 2) *justification:* an explicit description of the rationale behind each inferential step; 3) *strategy:* a high-level goal structure determining the problem-solving strategy used. From this typology, the authors claim that –through their empirical analysis– the most useful type of explanation is "justification". We contend that the type of explanations we propose correspond to both the "justification" and the "strategy" types; that is, we are giving not only the strategy used by the system to achieve the conclusion, but also the rationale behind each argument, which is clearly stated by its role in the dialectical tree.

We agree with [4], in that *"argumentation and explanation facilities in knowledge-base systems should be investigated in conjunction"*. Therefore, we propose a type of explanation that attempts to fill the gap in the area of explanations in argument systems. Our approach is to provide a higher-level explanation in a way that the whole context of a query can be revealed. The examples given in this paper stress this point.

This paper is organized as follows: first we will briefly outline the DELP concepts, then we will introduce δ-Explanations and their relation with DELP's answers, and finally we will discuss the related literature.

2 DeLP Overview

Defeasible Logic Programming (DELP) combines results of Logic Programming and Defeasible Argumentation. The system is fully implemented and available online [6]. A brief explanation is included below (see [7] for full details). It has the declarative capability of representing weak information in the form of *defeasible rules*, and a defeasible argumentation inference mechanism for warranting the entailed conclusions. A DELP-program $\mathcal{P}$ is a set of facts, strict rules and defeasible rules defined as follows. Facts are ground literals representing atomic information or the negation of atomic information using the strong negation "$\sim$" (*e.g.*, $chicken(little)$ or $\sim scared(little)$). *Strict Rules* represent non-defeasible information and are denoted $L_0 \leftarrow L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground literals (*e.g.*, $bird \leftarrow chicken$) or $\sim innocent \leftarrow guilty$). *Defeasible Rules* represent tentative information and are denoted $L_0 -\!\prec L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground literals. (*e.g.*, $\sim flies -\!\prec chicken$ or $flies -\!\prec chicken, scared$).

When required, $\mathcal{P}$ is denoted (Π, Δ) distinguishing the subset Π of facts and strict rules, and the subset Δ of defeasible rules (see Example 1). *Strong negation* is allowed in the head of rules, and hence may be used to represent contradictory knowledge. From a program (Π, Δ) contradictory literals could be derived. Nevertheless, the set Π (which is used to represent non-defeasible information) must possess certain internal coherence. Therefore, no pair of contradictory literals can be derived from Π.

A defeasible rule is used to represent tentative information that may be used if nothing could be posed against it. Observe that strict and defeasible rules are ground. However, following the usual convention [8], some examples use "schematic rules" with variables. To distinguish variables, as usual, they start with an uppercase letter.

Example 1. *Consider the* DELP*-program* (Π_1, Δ_1) *where:*

$$\Pi_1 = \left\{ \begin{array}{ll} (bird(X) \leftarrow chicken(X)) & chicken(little) \\ chicken(tina) & bird(rob) \\ scared(tina) & \end{array} \right\}$$

$$\Delta_1 = \left\{ \begin{array}{l} flies(X) -\!\prec bird(X) \\ flies(X) -\!\prec chicken(X), scared(X) \\ \sim flies(X) -\!\prec chicken(X) \end{array} \right\}$$

This program has three defeasible rules representing tentative information about the flying ability of birds in general, and about regular chickens and scared ones. It also has a strict rule expressing that every chicken is a bird, and three facts: 'tina' and 'little' are chickens, and 'tina' is scared.

From a program is possible to derive contradictory literals, *e.g.*, from (Π_1, Δ_1) of Example 1 it is possible to derive $flies(tina)$ and $\sim flies(tina)$. For the treatment of contradictory knowledge DELP incorporates a defeasible argumentation formalism. This formalism allows the identification of the pieces of knowledge that are in contradiction, and a *dialectical process* is used for deciding which information prevails as warranted. This dialectical process (see below) involves the construction and evaluation of arguments that either support or interfere with the query under analysis. Once the analysis is done, the generated arguments will represent *an explanation* for the query. As we will show next, arguments that explain an answer for a given query will be shown in a particular way using *dialectical trees*. The definition of dialectical tree will be included below, but first, we will give a brief explanation of other related concepts (for the details see [7]).

Definition 1 (Argument Structure). *Let* (Π, Δ) *be a* DELP*-program,* $\langle \mathcal{A}, L \rangle$ *is an argument structure for a literal* L *from* (Π, Δ)*, if* $\mathcal{A}$ *is the minimal set of defeasible rules* ($\mathcal{A} \subseteq \Delta$)*, such that: (1) there exists a defeasible derivation for* L *from* $\Pi \cup \mathcal{A}$*, and (2) the set* $\Pi \cup \mathcal{A}$ *is non-contradictory.*

Example 2. *From the* DELP*-program* (Π_1, Δ_1) *the following arguments can be obtained (due to space restrictions 'tina' will be abbreviated to 't' and 'flies(tina)' to 'f'):*
$\langle \mathcal{A}_1, f \rangle = \langle \{flies(t) -\!\prec bird(t)\}, flies(t) \rangle$
$\langle \mathcal{A}_2, \sim f \rangle = \langle \{\sim flies(t) -\!\prec chicken(t)\}, \sim flies(t) \rangle$
$\langle \mathcal{A}_3, f \rangle = \langle \{flies(t) -\!\prec chicken(t), scared(t)\}, flies(t) \rangle$

In DELP a literal L is *warranted* if there exists a non-defeated argument $\mathcal{A}$ supporting L. To establish if $\langle \mathcal{A}, L \rangle$ is a non-defeated argument, *defeaters* for $\langle \mathcal{A}, L \rangle$ are

considered, *i.e.*, counter-arguments that by some criterion are preferred to $\langle \mathcal{A}, L\rangle$. It is important to note that in DELP the argument comparison criterion is modular and thus, the most appropriate criterion for the domain that is being represented can be selected. In the examples in this paper we will use *generalized specificity* [9], a criterion that favors two aspects in an argument: it prefers (1) a *more precise* argument (*i.e.*, with greater information content) or (2) a *more concise* argument (*i.e.*, with less use of rules). Using this criterion in Ex. 2 $\langle \mathcal{A}_3, f\rangle$ is preferred to $\langle \mathcal{A}_2, \sim f\rangle$ (more precise) and $\langle \mathcal{A}_2, \sim f\rangle$ is preferred to $\langle \mathcal{A}_1, f\rangle$ (the later use the strict rule $bird(X) \leftarrow chicken(X)$).

A defeater $\mathcal{D}$ for an argument $\mathcal{A}$ can be *proper* ($\mathcal{D}$ is preferred to $\mathcal{A}$) or *blocking* (same strength). Since defeaters are arguments, there may exist defeaters for them, and defeaters for these defeaters, and so on. Thus, a sequence of arguments called *argumentation line* is constructed, where each argument defeats its predecessor. To avoid undesirable sequences, that may represent circular or fallacious argumentation lines, in DELP an argumentation line has to be *acceptable*, that is, it has to be finite, an argument can not appear twice, and supporting arguments, *i.e.*, in odd positions, (resp. interfering arguments) have to be not contradictory (see [7]).

Example 3. *(Extends Ex. 2) The argument $\langle \mathcal{A}_2, \sim f\rangle$ is a proper defeater of $\langle \mathcal{A}_1, f\rangle$, and $\langle \mathcal{A}_3, f\rangle$ is a proper defeater of $\langle \mathcal{A}_2, \sim f\rangle$. Hence, $[\langle \mathcal{A}_1, f\rangle, \langle \mathcal{A}_2, \sim f\rangle, \langle \mathcal{A}_3, f\rangle]$ is an acceptable argumentation line.*

Clearly, there can be more than one defeater for a particular argument $\mathcal{A}$. Therefore, many acceptable argumentation lines could arise from $\mathcal{A}$, leading to a tree structure. Given an argument $\langle \mathcal{A}_0, h_0\rangle$, a *dialectical tree* [7] for $\langle \mathcal{A}_0, h_0\rangle$, denoted $\mathcal{T}(\langle \mathcal{A}_0, h_0\rangle)$, is a tree where every node is an argument. The root of $\mathcal{T}(\langle \mathcal{A}_0, h_0\rangle)$ is $\langle \mathcal{A}_0, h_0\rangle$, and every inner node is a defeater (proper or blocking) of its parent. Leaves correspond to non-defeated arguments. In a dialectical tree every path from the root to a leaf corresponds to a different acceptable argumentation line. Thus, a dialectical tree provides a structure for considering all the possible acceptable argumentation lines that can be generated for deciding whether an argument is defeated. We call this tree *dialectical* because it represents an exhaustive dialectical analysis for the argument in its root.

Given a literal h and an argument $\langle \mathcal{A}, h\rangle$ to decide whether a literal h is warranted, every node in the dialectical tree $\mathcal{T}(\langle \mathcal{A}, h\rangle)$ is recursively marked as "D" (*defeated*) or "U" (*undefeated*), obtaining a marked dialectical tree $\mathcal{T}^*(\langle \mathcal{A}, h\rangle)$. Nodes are marked by a bottom-up procedure that starts marking all leaves in $\mathcal{T}^*(\langle \mathcal{A}, h\rangle)$ as "U"s. Then, for each inner node $\langle \mathcal{B}, q\rangle$ of $\mathcal{T}^*(\langle \mathcal{A}, h\rangle)$, (a) $\langle \mathcal{B}, q\rangle$ will be marked as "U" iff every child of $\langle \mathcal{B}, q\rangle$ is marked as "D", or (b) $\langle \mathcal{B}, q\rangle$ will be marked as "D" iff it has at least a child marked as "U".

Given an argument $\langle \mathcal{A}, h\rangle$ obtained from $\mathcal{P}$, if the root of $\mathcal{T}^*(\langle \mathcal{A}, h\rangle)$ is marked as "U", then we will say that $\mathcal{T}^*(\langle \mathcal{A}, h\rangle)$ *warrants* h and that h is *warranted* from $\mathcal{P}$.

In this paper, marked dialectical trees will be depicted as a tree of labelled triangles where edges denote the defeat relation (in Figure 1 three marked dialectical trees are shown). A double arrow edge represents a blocking defeat, whereas a single arrow represents a proper defeat. An argument $\langle \mathcal{A}, h\rangle$ will be depicted as a triangle, where its upper vertex is labelled with the conclusion h, and the set of defeasible rules $\mathcal{A}$ are associated with the triangle itself. At the right of each node the associated mark ("U" or "D") will be shown.

Example 4. *(Extends Ex. 3) Figure 1 shows the marked dialectical tree for* $\mathcal{T}^*(\langle \mathcal{A}_1, f\rangle)$ *(the leftmost tree), which has only one argumentation line. Observe that the argument* $\langle \mathcal{A}_2, \sim f\rangle$ *interferes with the warrant of 'flies(tina)' and the argument* $\langle \mathcal{A}_3, f\rangle$ *reinstates* $\langle \mathcal{A}_1, f\rangle$*. The root of* $\mathcal{T}^*(\langle \mathcal{A}_1, f\rangle)$ *is marked as "U" and therefore the literal 'flies(tina)' is warranted.*

3 DeLP Answers and δ-Explanations

Next, we will define *queries*, *answers* and *explanations*. We will introduce two types of queries: ground (called DELP-queries) and schematic. For both types of queries we will define explanations and a way to obtain the corresponding answer, that is: YES, NO, UNDECIDED or UNKNOWN.

Definition 2 (Queries). *A* DELP-query *is a ground literal that* DELP *will try to warrant. A query with at least one variable will be called* schematic query *and will represent the set of* DELP*-queries that unify with the schematic one.*

The dialectical process for warranting a query involves the construction and evaluation of several arguments that either support or interfere with the query under analysis. These generated arguments are connected through the defeat relation and are organized in dialectical trees. Observe that given a query Q there could exist different arguments that support Q, and each argument will generate a different dialectical tree. Therefore, as we will show below, the returned answer for Q will be only 'the tip of the iceberg' of a set of several dialectical trees that have been explored to support the resulting answer.

Thus, to understand why a query has a particular answer, it is essential to consider which arguments have been generated and what connections exist among them. In DELP, δ*-Explanations* for answers will be the set of dialectical trees that have been explored to obtain a warrant for that query. The definition for a δ-Explanation for a DELP-query follows, whereas explanations for schematic queries will be introduced by the end of this Section.

3.1 δ-Explanations for DELP-Queries

We contend that δ-Explanations are a central part of an argumentation system whose proof procedure is based on dialectical trees, because they allow to visualize the reasoning carried out by the system, and the support for the answer. It is clear that without this information at hand it will be very difficult to understand the returned answer. Next, we will introduce explanations for ground queries. Then, we will generalize explanations for schematic queries. Given a literal L, the complement with respect to strong negation will be denoted $\overline{L}$ (*i.e.*, $\overline{a}$=$\sim a$ and $\overline{\sim a}$=a).

Definition 3 (δ-Explanation).
Let $\mathcal{P}$ *be a* DELP*-program and* Q *a* DELP*-query. Let* $\langle \mathcal{A}_0, Q\rangle$,...,$\langle \mathcal{A}_n, Q\rangle$ *be all the arguments for* Q *from* $\mathcal{P}$*, and* $\langle \mathcal{B}_0, \overline{Q}\rangle$,...,$\langle \mathcal{B}_m, \overline{Q}\rangle$ *be all the arguments for* $\overline{Q}$ *from* $\mathcal{P}$*. Then, the* explanation *for* Q *in* $\mathcal{P}$ *is the set of marked dialectical trees* $\mathcal{E}_{\mathcal{P}}(Q) = \{\mathcal{T}^*(\langle \mathcal{A}_0, Q\rangle),\ldots,\mathcal{T}^*(\langle \mathcal{A}_n, Q\rangle)\} \cup \{\mathcal{T}^*(\langle \mathcal{B}_0, \overline{Q}\rangle),\ldots,\mathcal{T}^*(\langle \mathcal{B}_m, \overline{Q}\rangle)\}$.

Now it is possible to define DELP-answers in terms of their δ-Explanation.

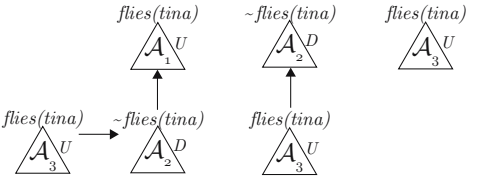


Fig. 1. δ-Explanation for $flies(tina)$

Definition 4 (DELP**-answer).** *Given a* DELP*-program $\mathcal{P}$ and a* DELP*-query Q, the answer for Q is either:*

YES, *if at least one tree in $\mathcal{E}_{\mathcal{P}}(Q)$ warrants Q.*
NO, *if at least one tree in $\mathcal{E}_{\mathcal{P}}(Q)$ warrants $\overline{Q}$.*
UNDECIDED, *if no tree in $\mathcal{E}_{\mathcal{P}}(Q)$ warrants Q nor $\overline{Q}$.*
UNKNOWN, *if Q is not in the signature of $\mathcal{P}$.*

Example 5. *(Extends Ex. 4) Figure 1 shows the δ-Explanation for the* DELP*-query '$flies(tina)$', where two dialectical trees for '$flies(tina)$' are marked "U". Therefore, '$flies(tina)$' is warranted and the answer is* YES*. Note that the δ-Explanation of Figure 1 is also an explanation for query '$\sim flies(tina)$' which answer is* NO*. Finally, observe that the answer for 'walks(tim)' is* UNKNOWN*, because it is not in the program signature.*

Remark 1. *The explanation for complementary literals will always be the same, since it is composed by both the trees for the literal and the trees for its complement.*

As we will show in the examples below, the semantics of the programs is *sensitive* to the addition or deletion of rules and facts. That is, a new fact added to a program can have a big impact on the number of arguments that can be built from the modified program. Taking into account this characteristic and considering the many possible interactions among arguments via the defeat relation (that lead to the construction of different dialectical trees), δ-Explanations become essential for understanding the reasons that support an answer.

Example 6. *Consider the* DELP*-program (Π_6, Δ_6):*

$$\Pi_6 = \{q, t\}\ \Delta_6 = \left\{ \begin{array}{l} (r \prec q)\ (\sim r \prec q, s) \\ (r \prec s)\ (\sim r \prec t) \end{array} \right\}$$

where the following arguments can be built:
$\langle \mathcal{R}_1, \sim r \rangle = \langle \{\sim r \prec t\}, \sim r \rangle \quad \langle \mathcal{R}_2, r \rangle = \langle \{r \prec q\}, r \rangle$
From this program the answer for the query 'r' is UNDECIDED*, and Figure 2 shows its δ-Explanation. Note that, although the literal 's' is in the program signature (in the body of a rule), there is no supporting argument for it. Therefore, the answer for query 's' is* UNDECIDED*, and the δ-Explanation is the empty set (i.e., $\mathcal{E}_{(\Pi_6, \Delta_6)}(s) = \emptyset$).*

Remark 2. DELP*-queries with* UNKNOWN *answers always have an empty δ-Explanation. However,* DELP*-queries that have* UNDECIDED *answers may have empty or non-empty explanations. Finally,* DELP*-queries with* YES *or* NO *answers will always have a non-empty explanation.*

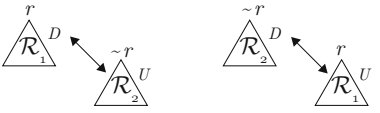


Fig. 2. δ-Explanation $\mathcal{E}_{(\Pi_6,\Delta_6)}(r)$

Example 7 shows how the introduction of a single fact in (Π_6, Δ_6) makes a significant difference in $\mathcal{E}_{(\Pi_6,\Delta_6)}(r)$.

Example 7. *(Extends Ex. 6) Consider the* DELP*-program* $(\Pi_6 \cup \{s\}, \Delta_6)$ *where the fact 's' is added to the program of Example 6. If we query for 'r' again, we get the answer* NO *with the* δ*-Explanation shown in Figure 3. Note that this* δ*-Explanation consists now of two more trees than the one in the previous example. This is so because there are two newly generated arguments:*

$\langle \mathcal{R}_3, r\rangle = \langle \{r \prec s\}, r\rangle \quad \langle \mathcal{R}_4, \sim r\rangle = \langle \{\sim r \prec q, s\}, \sim r\rangle$

It is our contention that, in DELP, the answer for a query can be easily explained by presenting the user the associated dialectical trees. From this set of trees the answer becomes thoroughly justified, and the context of the query is revealed. The following examples have more elaborated DELP-programs and the δ-Explanations show that a defeater $\mathcal{D}$ for $\mathcal{A}$ may attack an inner point of $\mathcal{A}$.

Example 8. *Consider the* DELP*-program* (Π_8, Δ_8)*:*

$$\Delta_8 = \left\{ \begin{array}{ll} (a \prec b) & (b \prec c) \\ (\sim b \prec d) & (d \prec e) \\ (\sim d \prec f, e) & (\sim b \prec e) \\ (a \prec x) & (x \prec c) \\ (\sim x \prec e) & (a \prec h) \\ (h \prec f) & (\sim h \prec i) \end{array} \right\} \quad \Pi_8 = \{c, e, f\}$$

where the following arguments can be built:

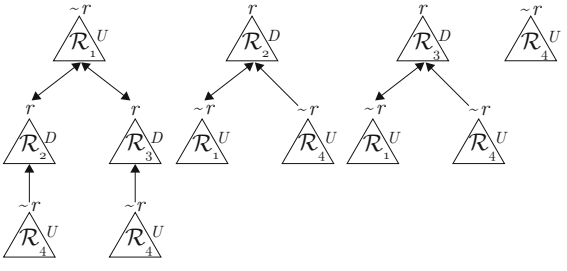


Fig. 3. δ-Explanation $\mathcal{E}_{(\Pi_6\cup\{s\},\Delta_6)}(r)$

$\langle \mathcal{B}_1, b\rangle = \langle \{b \prec c\}, b\rangle$ $\quad$ $\langle \mathcal{B}_2, \sim b\rangle = \langle \{\sim b \prec e\}, \sim b\rangle$
$\langle \mathcal{X}_1, x\rangle = \langle \{x \prec c\}, x\rangle$ $\quad$ $\langle \mathcal{X}_2, \sim x\rangle = \langle \{\sim x \prec f\}, \sim x\rangle$
$\langle \mathcal{D}_1, d\rangle = \langle \{d \prec e\}, d\rangle$ $\quad$ $\langle \mathcal{D}_2, \sim d\rangle = \langle \{(\sim d \prec f, e)\}, \sim d\rangle$
$\langle \mathcal{A}_1, a\rangle = \langle \{(a \prec h), (h \prec f)\}, a\rangle$

From (Π_8, Δ_8) the answer for 'a' is YES*, and the answer for '$\sim a$' is* NO*. As stated in Remark 1, although both queries have different answers, they both have the same δ-Explanation, which is depicted in Figure 4. In that figure, sub-arguments are represented as smaller triangles contained in the triangle which corresponds to the main argument at issue. For instance, the argument $\langle \mathcal{B}_2, \sim b\rangle$ defeats $\langle \mathcal{B}_1, b\rangle$ that is a subargument of $\langle \{(a \prec b), (b \prec c)\}, a\rangle$.*

Example 9. *Consider the* DELP*-program $(\Pi_8 \cup \{i\}, \Delta_8)$ where the fact 'i' is added to the program of Example 8. Now the argument $\langle \mathcal{H}_2, \sim h\rangle$ can be generated which is a defeater for $\langle \mathcal{H}_1, h\rangle$ (a subargument of $\langle \mathcal{A}_1, a\rangle$):*
$\langle \mathcal{H}_2, \sim h\rangle = \langle \{\sim h \prec i\}, \sim h\rangle$ $\quad$ $\langle \mathcal{H}_1, h\rangle = \langle \{h \prec f\}, h\rangle$
Here, argument $\mathcal{H}_2$ blocks argument $\mathcal{H}_1$ (subargument of $\mathcal{A}_1$), leaving no undefeated arguments for 'a'; then, the answer for both 'a' and '$\sim a$' is UNDECIDED*. The rest of the explanation remains the same as the one in Figure 4.*

From the DELP programmer point-of-view, δ-Explanations give a global idea of the interactions among arguments within the context of a query. This is an essential debugging tool when programming: if unexpected behaviour arises, the programmer can check the given explanations to detect errors.

In the previous examples we have not shown an explanation associated with a query with an UNKNOWN answer, because this type of answers have an empty δ-Explanation. Finally, observe that queries that do not correspond to the intended domain of the program will return the answer UNKNOWN. This will capture errors like querying for "fly" instead of "flies", or a query like "penguin(X)" in Example 1.

3.2 Explanations for Schematic Queries

A *schematic query* is a query that has at least one variable (see Definition 2), and hence, it represents the set of DELP-queries that unify with it. Now, we will extend the definition of δ-Explanation to include schematic queries. Consider again the DELP-program

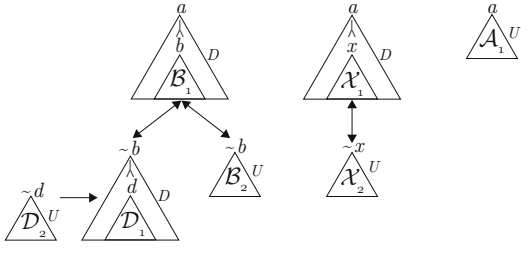


Fig. 4. δ-Explanation $\mathcal{E}_{(\Pi_8, \Delta_8)}(a)$

of Example 1, the schematic query $flies(X)$ has actually infinite terms that unify with variable X. However, all queries with terms that are not in the program signature will produce an UNKNOWN answer and therefore an empty explanation. Thus, the set of instances of a schematic query that will be considered for generating an explanation will refer only to those instances of DELP-queries that contain constants from the program signature. An explanation for a schematic query will be the set of δ-Explanations of those instances whose answers are YES, NO, or UNDECIDED.

Definition 5 (Generalized δ-Explanation).
Let $\mathcal{P}$ be a DELP*-program and Q a schematic query. Let $\{Q_1, \ldots, Q_z\}$ be all the instances of Q so that their* DELP*-answer is different from* UNKNOWN*. Let $\mathcal{E}_\mathcal{P}(Q_i)$ be the δ-Explanation for the* DELP*-query Q_i ($1 \leq i \leq z$) from program $\mathcal{P}$. Then, the* generalized δ-Explanation *for Q in $\mathcal{P}$ is $\mathcal{E}_\mathcal{P}(Q) = \{ \mathcal{E}_\mathcal{P}(Q_1), \ldots, \mathcal{E}_\mathcal{P}(Q_z)\}$.*

Observe that a δ-Explanation (Definition 3) is a particular case of a Generalized δ-Explanation, where the set $\mathcal{E}_\mathcal{P}(Q)$ is a singleton.

Example 10. *Consider again the* DELP*-program (Π_1, Δ_1), and suppose that we want to know if from this program it can be warranted that a certain individual does not fly. If we query for $\sim flies(X)$, the answer is* YES*, because there is a warranted instance: $\sim flies(little)$. The supporting argument is ('little' was abbreviated to 'l'):*
$\langle \mathcal{B}_1, \sim flies(l) \rangle = \langle \{\sim flies(l) \prec chicken(l)\}, \sim flies(l) \rangle$
The trees of the generalized explanation are shown in Figure 5. This explanation also shows that the other instance ($\sim flies(tina)$) is not warranted.

It is important to note that the answer for the schematic query $flies(X)$ is also YES*, but with a different set of warranted instances: $flies(tina)$ and $flies(rob)$. The supporting argument for instance '$X = tina$' was already discussed, and the undefeated argument for instance '$X = rob$' is:*
$\langle \mathcal{C}_1, flies(rob) \rangle = \langle \{flies(rob) \prec bird(rob)\}, flies(rob) \rangle$

The generalized δ-Explanation for $flies(X)$ is the same as the one for $\sim flies(X)$, depicted in Figure 5 (see Remark 1).

Definition 6 (DELP**-answer for a schematic query).** *Given a* DELP*-program $\mathcal{P}$ and a schematic query Q, the answer for Q is*

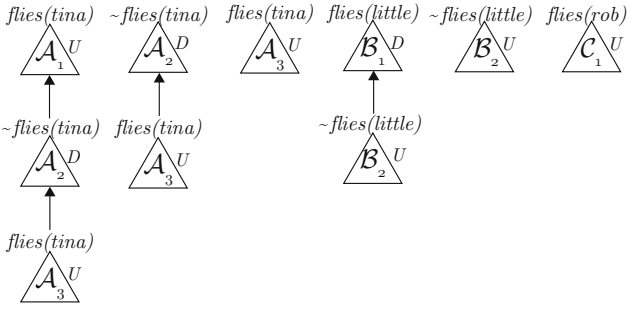


Fig. 5. Generalized δ-Explanation for '$\sim flies(X)$'

- YES, *if there exists an instance Q_i of Q such that at least one tree in $\mathcal{E}_{\mathcal{P}}(Q_i)$ warrants Q_i.*
- NO, *if for every instance Q_i of Q that is in the signature of $\mathcal{P}$, there is no tree in $\mathcal{E}_{\mathcal{P}}(Q_i)$ that warrants Q_i, and there exists an instance $\overline{Q_i}$ of $\overline{Q}$ such that at least one tree in $\mathcal{E}_{\mathcal{P}}(\overline{Q_i})$ warrants $\overline{Q_i}$.*
- UNDECIDED, *if for every instance Q_i of Q that is in the signature of $\mathcal{P}$, there is no tree in $\mathcal{E}_{\mathcal{P}}(Q_i)$ that warrants Q_i nor $\overline{Q_i}$.*
- UNKNOWN, *if there is no instance Q_i of Q such that Q_i is in the signature of $\mathcal{P}$.*

Observe that Definition 4 is a particular case of the previous definition, where there is a single instance of Q.

Example 11. *Consider the following* DELP*-program:*

$$\Pi_{11} = \left\{ \begin{array}{ll} adult(peter) & adult(annie) \\ unemployed(peter) & student(annie) \end{array} \right\}$$

$$\Delta_{11} = \left\{ \begin{array}{l} has\_a\_car(X) -\!\prec adult(X) \\ \sim has\_a\_car(X) -\!\prec unemployed(X) \\ \sim has\_a\_car(X) -\!\prec student(X) \end{array} \right\}$$

where the following arguments can be built('has_a_car' was replaced by 'car', '$annie$' by 'a', and '$peter$' by 'p'):
$\langle \mathcal{A}_1, car(a) \rangle$ = $\langle \{car(a) -\!\prec adult(a)\}, car(a) \rangle$
$\langle \mathcal{A}_2, \sim car(a) \rangle$ = $\langle \{\sim car(a) -\!\prec student(a)\}, \sim car(a) \rangle$
$\langle \mathcal{P}_1, car(p) \rangle$ = $\langle \{car(p) -\!\prec adult(p)\}, car(p) \rangle$
$\langle \mathcal{P}_2, \sim car(p) \rangle$ = $\langle \{\sim car(p) -\!\prec unemployed(p)\}, \sim car(p) \rangle$

When querying for '$has\_a\_car(X)$', variable 'X' unifies with both '$annie$' and '$peter$'. Then, DELP *builds arguments for both instances: $\mathcal{A}_1$ and $\mathcal{A}_2$ for '$X = annie$', and $\mathcal{P}_1$ and $\mathcal{P}_2$ for '$X = peter$'. From Figure 6, it is clear that no argument is undefeated, i.e., there is no tree that warrants '$has\_a\_car(X)$', for either of the two instances. Therefore, the answer is* UNDECIDED*, and the variable remains unbound.*

Schematic queries give us the possibility of asking more general questions than ground queries. Now we are not asking whether a certain piece of knowledge can be believed, but we are asking if there exists an instance of that piece of knowledge (related to an individual) that can be warranted in the system. This could lead to deeper reasoning as we may pose a query, gather the warranted instances and continue reasoning with those individuals.

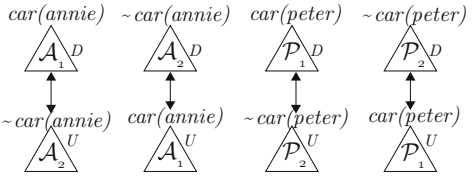


Fig. 6. Generalized δ-Explanation for '$has\_a\_car(X)$'

The δ-Explanations system receives a DELP-program P, a query Q and an argument comparison criterion C, and returns a δ-Explanation EX along with the proper answer ANS. The system is described by the following algorithm in a Prolog-like notation:

```
d_Explanations(P,C,Q,EX,ANS):- warrants(P,C,Q,WSQ),
             complement(Q,NQ), warrants(P,C,NQ,WSNQ),
             get_trees(WSQ,WSNQ,EX), get_answer(Q,WSQ,WSNQ,ANS).
warrants(Q,WS):- findall((Q,TREES),warrant(Q,TREES),WS).
get_answer(_,WSQ,WSNQ,yes):- WSQ \= [].
get_answer(_,WSQ,WSNQ,no):- WSNQ \= [].
get_answer(Q,_,_,unknown):- not_in_signature(Q).
get_answer(_,_,_,undecided).
```

Predicate warrant/2 takes a query and attempts to warrant it; it does so by building dialectical trees. In case the query is warranted, the dialectical trees built are 'saved' along with the query. Different instances of a query can be obtained via backtracking. Predicate warrants/2 takes a query Q and returns all its warranted instances (along with their corresponding trees) within a list. Predicate get_trees/3 retrieves the dialectical trees information from the warranted instances for both Q and $\sim$Q. Finally, predicate get_answer/4 takes the query, both lists of warranted instances (for Q and $\sim$Q), and returns the answer.

The above described system is fully implemented and offers support for queries, answers and explanations. Explanations are written into an XML file, which is parsed by a visualization applet. The visualization of trees belonging to dialectical explanations is enhanced by allowing the user to zoom-in/out, implode/explode arguments, *etc*. The internal structure of an argument is hidden when imploding, and a unique tag is shown instead.

Lemma 1 (δ-Explanation Soundness). *Let $\mathcal{P}$ be a* DELP*-program, C an argument comparison criterion, and Q a schematic query posed to $\mathcal{P}$. Let E be the δ-Explanation returned in support of the answer A. Then E justifies (Definition 6) A.*

Lemma 2 (δ-Explanation Completeness). *Let $\mathcal{P}$ be a* DELP*-program, C an argument comparison criterion, and Q a schematic query posed to $\mathcal{P}$. Let E be the δ-Explanation returned in support of the answer A. Then E contains all the possible justifications (Definition 6) for any instance of A.*

4 Related Work

A very thorough survey relating explanation and argumentation capabilities can be found in [4]. Although the authors are mainly concerned about negotiation/persuasion, and interactive/collaborative explanations, the discussion Section of that article poses really interesting issues about the integration of explanation and argumentation; for instance, whether the same knowledge base can be used to generate both explanatory and argumentative information. In our approach, we do extract all the information from the given knowledge base (*i.e.*, the DELP-program) to return both kinds of information.

In [4], the authors claim that these two areas (*i.e.*, argumentation and explanation facilities in knowledge-base systems) should be "investigated in conjunction". Our paper tries to move forward in that direction, providing means to "better understand the mechanisms underlying the activities of explanation and argumentation".

Recently, Douglas Walton [10] has offered a dialogue theory of explanation. In that work a successful explanation is defined as transfer of understanding in a dialogue system where a questioner and a respondent take part. The questioner begins by asking a question seeking to understand some piece of information and the respondent gives a reply that conveys understanding of that information to the questioner. His approach follows a different path than ours, focussing in the distinction between *explanation* and *argument* and defining an explanation as a new speech act.

Our approach handles δ-Explanations within argumentation systems through a graphical representation of dialectical trees. Visualization in argumentation has been addressed in [11]. In that paper, the objective is to provide a visual tool that does not require the reader to understand logic to be able to follow the argumentative process shown by the system. To achieve this, they use an animated argumentation space: arguments are introduced one by one in the process to allow for a more comprehensive visualization. They also allow to see this space in a static manner. Both ways give the user the possibility to navigate the space at will, or in auto-pilot mode. Every element taking part of the argumentation process is represented graphically: conflicts are highlighted and arguments are tagged with the role they are playing in the whole process.

Although the article by Schroeder uses argumentation trees in a similar way as we do, we focus on explanations; that is, we are concerned with providing the whole context corresponding to the query. Our explanations are represented in such a way that they are useful to both humans and software agents.

5 Conclusions and Future Work

Future work includes further research about additional information that can be attached to the current form of the δ-Explanations. In particular, we are currently formalizing the notion of *discarded arguments*. These arguments are discarded by the system in the sense that their introduction into an acceptable argumentation line renders it fallacious. At the moment, we have singled out two reasons for an argument $\mathcal{A}$ to be discarded: (1) Non-attacking arguments: when $\mathcal{A}$ conflicts with the last argument in the line, but does not attack it (*i.e.*, the last argument is better than $\mathcal{A}$ wrt. the comparison criterion); (2) Double-blocking arguments: when the final argument in the line $\mathcal{A}_n$ is a blocking defeater of the preceding argument $\mathcal{A}_{n-1}$, and $\mathcal{A}$ is, in turn, a blocking defeater for $\mathcal{A}_n$. More dialectical constraints can be considered thus adding more types of discarded arguments. It is interesting to show discarded arguments within a δ-Explanation, because the user has the possibility of analyzing why a particular argument has not been included into the explanation. Sometimes, it is not clear why these situations occur.

We have addressed the problem, not often considered, of providing explanation capabilities to an argumentation system. We have defined a concrete argument system with explanation facilities. We consider the structures that provide information on the

warrant status of a literal. As the system has been implemented, we are developing applications that uses the δ-Explanation system as subsystem.

References

1. Lacave, C., Diez, F.J.: A review of explanation methods for heuristic expert systems. Knowl. Eng. Rev. 19(2), 133–146 (2004)
2. Ye, L.R., Johnson, P.E.: The impact of explanation facilities on user acceptance of expert systems advice. MIS Q. 19(2), 157–172 (1995)
3. Guida, G., Zanella, M.: Bridging the gap between users and complex decision support systems: the role of justification. In: ICECCS '97: Proc. 3rd IEEE International Conference on Engineering of Complex Computer Systems, Washington, pp. 229–238. IEEE Computer Society Press, Los Alamitos (1997)
4. Moulin, B., Irandoust, H., Bélanger, M., Desbordes, G.: Explanation and argumentation capabilities: Towards the creation of more persuasive agents. Artif. Intell. Rev. 17(3), 169–222 (2002)
5. Falappa, M.A., Kern-Isberner, G., Simari, G.R.: Explanations, belief revision and defeasible reasoning. Artif. Intell. 141(1), 1–28 (2002)
6. DeLP: `http://lidia.cs.uns.edu.ar/delp`
7. García, A.J., Simari, G.R.: Defeasible logic programming: An argumentative approach. Theory and Practice of Logic Programming 4(1), 95–138 (2004)
8. Lifschitz, V.: Foundations of logic programs. In: Brewka, G. (ed.) Principles of Knowledge Representation, pp. 69–128. CSLI Pub. (1996)
9. Stolzenburg, F., García, A., Chesñevar, C.I., Simari, G.R.: Computing Generalized Specificity. Journal of Non-Classical Logics 13(1), 87–113 (2003)
10. Walton, D.: A new dialectical theory of explanation. Philosophical Explorations 7(1), 71–89 (2004)
11. Schroeder, M.: Towards a visualization of arguing agents. Future Generation Computer System 17(1), 15–26 (2000)

Arguing over Actions That Involve Multiple Criteria: A Critical Review

W. Ouerdane, N. Maudet, and A. Tsoukias

LAMSADE, Univ. Paris-IX, Place du Marchal De Lattre de Tassigny,
75775 Paris Cedex 16
{wassila.ouerdane,maudet,tsoukias}@lamsade.dauphine.fr

Abstract. There has recently been many proposals to adopt an argumentative approach to decision-making. As the underlying assumptions made in these different approaches are not always clearly stated, we review these works, taking a more classical decision theory perspective, more precisely a multicriteria perspective. It appears that these approaches seem to have much to offer to decision models, because they allow a great expressivity in the specification of agents' preferences, because they naturally cater for partial specification of preferences, and because they make explicit many aspects that are usually somewhat hidden in decision models. On the other hand, the typically intrinsic evaluation used in these approaches is not always the most appropriate, and it is not always clear how the multicriteria feature is taken into account when it comes to aggregating several arguments that may potentially interact.

1 Introduction

Decision-support systems aim at helping the user to shape a problem situation, formulate a problem and possibly try to establish a viable solution to it. Under such a perspective decision aiding can be seen as the construction of the reasons for which an action is considered a "solution to a problem" rather than the solution itself [Tso07]. Indeed the problem of decisions *accountability* is almost as important as the decision itself. Decision support can therefore be seen as an activity aiming to construct arguments through which a decision maker will convince first herself and then other actors involved in a problem situation that "that action" is the best one (we are not going to discuss the rationality hypotheses about "best" here). Decision Theory and Multiple Criteria Decision Analysis have focussed on such issues for a long time, but more on how this "best solution" should be established and less on how a decision maker should be convinced about that (for exceptions on that see [BMP+00, BS02]).

More recently, in the field of artificial intelligence, argumentation has been put forward as a very general approach allowing to support different kinds of decision-making [BG96, PV02, Pol87, PJ98, AP06]. Typically, one will construct for each possible decision (alternative) a set of positive arguments, and a set of negative arguments. However, decision-makers do not simply *list* pro and cons: they exchange arguments, some of them interacting with others, attacking or

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 308–319, 2007.

reinstalling previous arguments put forward (by the other party, or sometimes indeed by itself). Distinguishing what eventually should count as acceptable arguments has been the study of numerous studies, and necessitates to value the arguments. Cayrol and Lagasquie-Schiex [CLS05] distinguish *intrinsic* valuation of arguments (without any consideration for the other arguments –for instance it may be based on the credibility of the source), and *interaction-based valuation* of arguments (simply resulting of the interactions between arguments –for instance some may be better supported than others, etc.). In the seminal work of Dung [Dun95], different semantics are proposed, which interpret differently what (sets of, in this case) arguments should be considered acceptable, only based on their interaction-based valuation. More recently, some approaches propose to take both aspects into account, see *e.g.* [KP98]. Once the valuation has been made, it is then possible to select the acceptable arguments. Usually, only a crisp selection is allowed: arguments are acceptable or not; however a more gradual acceptability is also possible [CLS05]. Our objective in this paper is to clarify the connections between argumentation and decision-making, and more precisely to inspect the recent proposals that have been put forward to handle (multi-criteria) decision-making in an argumentative framework.

The rest of this paper is as follows. In Section 2, we examine more carefully what it means to argue for an action, especially when different points of view can be considered to assess that action. We confront the different proposals put forward in the literature to our multicriteria perspective and discuss some hidden assumptions that they make. In Section 3, we move on to the following step by inspecting how aggregation can then be performed. Section 4 concludes.

2 Arguing over Actions

Before reviewing the literature on argument-based decision making (focusing especially on how they account for the fact that different criteria may be involved), we start by briefly recalling what makes decision over actions different from decision over beliefs.

2.1 Arguments Meet Beliefs, Actions, and "Points of View"

Argumentation is usually conceived as a process for handling (potentially conflicting) *beliefs*. In AI, many systems have been proposed that allow to capture the defeasible nature of this kind of reasoning. Under this perspective, the basic building block (the argument) can typically be defined as a premise/conclusion pair, whereby you state that this conclusion should be reached under these premises. What is discussed here is the truth-value of the conclusion, so an argument supporting a conclusion basically asserts some evidence to believe that this conclusion holds.

When it comes to decision-making though, this rather crude argument scheme needs to be refined. Indeed, as it has been recognized for a long-time now, a significant difference exists between argumentation for beliefs and argumentation

for actions [FP97, FP98]. This is best explained by means of a simple example, inspired by [FP97]. Saying that some symptoms "support" a given diagnosis, and that this diagnosis in turn "support" a given medication are two different things. The first —epistemic— argument is typically a defeasible proof of the doctor's diagnosis. The latter —practical— argument is a recommendation that this course of action should be chosen, which can for instance be defeated by the fact that other medications may turn out to be better options. So the same word "support" must be interpreted differently.

But we need to make more precise what is exactly meant by "an argument is in favour of an action a". The intuitive reading is that action a will have "good consequences". So we must first somehow *valuate* the outcome of the action. In decision models, this would typically be done by using an ordered scale defining the different values that can be used to assess the action (for instance, marks from 0 to 20 for students). Now what counts as a positive or negative outcome is specific to each agent, and depends of its (subjective) preferences. That is, you must classify the outcome of the actions. In decision models, one classical approach is that the agent uses an evaluation scale and specify a frontier, that is, a neutral point (or zone), thus inducing a *bipolar scale*. This will in turn allow us to determine what counts as an argument pro, or against, the action.

Let us suppose that we want to select a candidate for a given position, and that we have a number of candidates applying for it. We need to evaluate the outcome of each possible action, that is, how good is the situation induced by accepting each given candidate. For instance, a desired consequence is to have a strong enough candidate as far as academic level is concerned. Let us suppose that this is assessed by using a bipolar scale referring to marks, where 12 stands for our neutral point. Then, we could say that according to "marks", we have an argument in favour of accepting this candidate if its mark is more than 12.

Intuitively, as we said before, performing an action will bring about a state of the world which will be judged desirable or not. In general however, it is possible that you may have different valuations that you assign to a given action, depending on different points of view that you take to evaluate that action. Very often, these different valuations cannot be merged into a single point of view. This has been recognized in particular in multi-criteria decision-making, where a criterion is regarded as a point of view against which it is possible to compare different actions. Now, the definition of a neutral point for each point of view defines what we shall call here a *neutral action*, a special action against which each action can be compared.

2.2 Discussion of Existing Approaches

In [FP97], Fox and Parsons proposed one of the first account that tried to advocate an argumentative approach to decision-making, building on Fox's earlier work [FBB80]. They recognize and clearly state what makes argumentation for actions different from argumentation for beliefs, and put forward the following argument scheme:

Fox and Parsons' Argument Scheme
We should perform A (A has positive expected value)
Whose effects will lead to the condition C
Which has a positive value

As explained by Fox and Parsons, the advantage of this representation is that it makes explicit three inference steps: (i) that C will indeed result from action A, (ii) that C has some positive value, and eventually (iii) that A has a positive expected value. Clearly, steps (ii) and (iii) requires additional information in order to be able to assign values to situations, and to decide whether the action has indeed a positive expected value. The valuation of the condition is subjective (dependent of the agent's preference), and represented here by "labelling the proposition describing C with a sign drawn from a dictionary", which can be qualitative or not and plays the role of a scale. Interestingly, they also allow for different points of view over which values can be assigned.

So for instance, opting for a given candidate (say a) could lead to an outcome where the chosen candidate has a mark of 14 (this would be captured by the first epistemical step e_1 of the scheme, where ga stands for the justification of this step). Together with the two following steps, this could be represented with this scheme as follows:

$$\begin{array}{lll} chose_a \rightarrow mark = 14 : ga & : + & e_1 \\ mark = 14 : va & : + & v_1 \\ chose_a & : (e_1, v1) : + & ev_1 \end{array}$$

The last step concludes that this action has a positive expected value. More interestingly, the second step (v_1) means that the condition $mark = 14$ is positively evaluated by our agent (noted by symbol $+$) (it then counts as a positive argument), where va is the justification for this value assignment. Although this aspect is not deeply explored in the paper, a very interesting feature of this approach is then that it makes explicit the grounds allowing to assign this value to this condition: what may count ha obvious candidates to justify this value assignment, if we take the view of the multicriteria-decision approach, would be the user's preferences ("I consider that the mark is good from 12"), as well as the preference model used ("I consider this to be a positive argument as long as it is beyond the limit previously stated").

But we could also directly encode within this scheme that opting for a given candidate would lead to an outcome where the condition that the chosen candidate has a mark over 12 is satisfied, a fact that we consider positive. This could be represented as follows (the last step does not vary)

$$\begin{array}{ll} chose_a \rightarrow mark \geq 12 : ga : + & e_1 \\ mark \geq 12 : va : + & v_1 \end{array}$$

meaning that the condition $mark \geq 12$ is positively evaluated by our agent (noted by symbol $+$) (it then counts as a positive argument), where va is the justification for this value assignment. In this case, the nature of this justification is less clear, for it leads to support the agent's preferences.

These two alternative ways of representing argument schemes about actions seem somewhat unsatisfactory. On the one hand, chosing to directly represent the neutral action drawn from the agent's preferences drops the relation linking an action and its consequences. On the other hand, not representing it assumes it is somehow encoded within a "value assignment" mechanism. Finally, this approach does not really acknowledge that actions themselves can be evaluated against a number of meaningful, predefined, dimensions: in fact, each condition induces a new dimension against which the action can be evaluated.

One of the most convincing proposal recently put forward to account for argument-based decision-making is the one by Atkinson et al. [ABCM06, Atk06]. They propose an extension of the "sufficient condition" argument scheme proposed by Walton [Wal96].

Atkinson's Argument Scheme
In the circumstances R
We should perform A
Whose effects will result in state of affairs S
Which will realise a goal G
Which will promote some value V

To avoid confusion with the previous approach, we must first make clear that the notion of value is used here in a different sense. As we shall see, it plays a role comparable to that of a criteria in multi-criteria decision making. Atkinson explains [Atk05] that values should not be confused with goals as "they provide the actual reasons for which an agent wishes to achieve a goal". *Goals* refer to single values, but an action can bring about a *state of affairs* that satisfy many goals, hence affecting different *values*. So, unlike the previous one, this approach explicitly represents both action's consequences, and states actually desired by the agent (preferences). We believe this distinction remains important even if there is no discrepancy between observed and inferred states [BCP06]. Technically, a function *value* maps goals to pairs $\langle v, sign\rangle$ where $v \in V$, and *sign* belongs to the scale $\{+, -, =\}$ (but Modgil [Mod06] adds a notion of degree to which the value is promoted). For instance, using our running example, we could have

$$value(mark \geq 12) = \langle academic\_level, + \rangle$$

meaning that the value (criteria) academic quality is promoted when the mark is over 12.

In this approach, values clearly play the role of criteria. So it looks like specifying goals amounts to specifying a (potentially partial) neutral action. However, the declarative nature of goals allows for more flexible classifications than what we typically have in decision models[1]. For instance, it is possible to easily express that

$$value(age \geq 18 \wedge age \leq 32) = \langle youth, + \rangle$$

[1] Although some approaches try to overcome these limitations, see for instance [aC96].

the value "youth" is only promoted when the *age* falls between 18 and 32. It is also important to note that values are eventually assigned to state of affairs *via* goals. So the justification of value assignment to states is implicitly given by the fact that the goal is reached (or not). One potential problem is that it does not leave any other option if we were to provide some additional justification (for instance related to preference model used). We also refer to [BCP06] for a detailed discussion related to this scheme.

In [ABP05], Amgoud et al. propose an approach explicitly linking argumentation to multi-criteria decision-making. They see an argument as a 4-tuple $\langle S, x, c, g\rangle$ where

- S is the support of the argument,
- x is the conclusion of the argument (the action)
- c is the criterion which is evaluated for x,
- g is the goal and represents the way c is satisfied by x

It is required that S is consistent when we add the fact that the action x has taken place. Here, in a way that is reminiscent of the previous approach, each goal g is explicitly associated to a criterion by means of a propositional formula $g \rightarrow c$, although the possibility of having goals referring to different criteria is also mentioned. In this approach, unlike in [Atk05], the use of (bipolar) scale is explicitly mentioned: the goals will fall either on the negative or on the positive side. Their approach also allows for quantitative measure of how good are the attained goals. So for instance, we may specify that knowledge base has several strata

$$G_2^+ = \{mark \geq 16\};\ G_1^+ = \{16 > mark \geq 12\};\ G_1^- = \{mark < 12\}$$

which means that the marks are considered as "good" from 12, and even "very good" from 16, while it is unsufficient when it is below 12. This comes together with formulae of the form

$$mark \geq 16 \rightarrow academic\_level$$

which explicitly states that the goal G_2^+ affects the criteria "academic level". Now each decision will have some consequences, that will in turn fulfill some goals or not. It is then possible to identify arguments pro and cons a given decision x, by simply scanning the knowledge base and checking which positive (resp. negative) goals are satisfied by the occurrence of a given decision x.

In a very recent proposal, Morge and Mancarella [MM07] propose a multi-attribute argumentation framework for opinion explanation. Here, a main goal is split into sub-goals and so on. They make a distinction between *high level goals* ("abstract goals that reveal the user's need"), and *low-level goals* ("criteria for evaluating different alternatives"). As for the satisfaction of a goal by a given decision, this is explicitly stated by:

- *decision rules* of the form $R : g \leftarrow D, B_1, \ldots, B_n$ meaning that the goal g *can* be achieved by decision D, given that conditions $B_1, \ldots, B_n$ are satisfied.

– *goal rules* of the form $R : g \leftarrow g_1, \ldots, g_n$ meaning that the head of the rule is reached if the goals listed in the body are reached

The notion of priority between rules allows to refine decision rules, in order to make more complex aggregation. So for instance; if we were to specify that we would chose an alternative if it meets one out of two goals, we would specify that

$$R_0 : g_0 \leftarrow g_1, g_2$$
$$R_1 : g_0 \leftarrow g_1$$
$$R_2 : g_0 \leftarrow g_2$$

together with the preferential information that $R_0 \succeq \{R_1, R_2\}$

Now if we inspect what plays the role of a criteria in this approach, it is difficult to say. In fact, there is no notion properly corresponding to that of a criteria: there is no point of view against it is possible to compare alternatives. It would be tempting to say that there exists an implicit preference model stating that the decision-maker prefers to satisfy goals, rather than not. However it is deceptive. It could well be that we have the following preference ordering between rules:

$$R_0 : g_0 \leftarrow g_1, g_2$$
$$R_1 : g_0 \leftarrow \neg g_1$$
$$R_2 : g_0 \leftarrow \neg g_2$$

again with $R_0 \succeq \{R_1, R_2\}$

In that case, it is clearly not possible to evaluate on a single point of view. It is only possible to say that we would prefer an action satisfying *both* g_1 and g_2, rather than only $\neg g_1$ or g_2. Only when the set of rules exhibits a very specific structure is it possible to interpret goals as proper criteria. In general however, this approach is more expressive and cater for preference models where "coalitions" of criteria are considered, which makes the comparison more difficult.

2.3 Discussion

In the previous section we have discussed several approaches to argument-based decision-making. What we have seen is that each approach is rather marginally different from the other ones, but that, by making explicit different steps of the process, they focus on different aspects of the process. Fox and Parsons are the only ones to explicitly represent the justification of a value assignment, however, they do not fully explore this avenue; and hardwire the possibility of having different criteria. Atkinson makes this latter distinction clear, but on the other hand, do not cater for an explicit representation of all the justifications of the value assignment (this only rely on the logical satisfaction: a goal is reached or not, which justifies the value assignment). In this case, it is not possible to represent or indeed challenge the preference structures used. Amgoud *et al.* also rely on the logical satisfaction of goals to justify the value assignment, but the

goals are ordered in a way that indeed allows to refine the preference structure, to express various degrees of satisfaction of a goal. Still, this is directly encoded in the knowledge base and cannot be discussed in the process. Also, by using a bipolar scale, they constrain the syntax of goals and prevent themselves from using the full expressivity provided by the logic. Overall, it is important to emphasize that the definition of the argument scheme is of primary importance: by expliciting the inference steps of an argument, we also define what counts as valid "critical question", that is how arguments will interact with each others (how they can be attacked and so on).

There are, on the other hand, many similarities between these approaches. First, the evaluation is made possible by an explicit representation of the consequences of the action. By relying on logic to represent these states of affairs, it is more expressive than the ordered scale that is usually used in decision models. One further possibility that is offered by this representation is that profile may be only partially defined, whereas in decision models you would require each action to be evaluated on each different criteria.

The third, perhaps most striking similarity, is that they all rely on a method of *intrinsic evaluation*, and use more or less explicitly a neutral action. In decision models, on the other hand, the canonical case is the *pairwise evaluation*, that is, actions are evaluated against each others, and not against a neutral action. Although the use of neutral action can be justified, it has some consequences and drawbacks that, we feel, is important to emphasize:

- the adoption of a neutral action makes very important the definition of each *neutral point*, that is, the frontier (or more generally zone). In particular, a seemingly insignificant modification of the frontier can have tremendous consequences (as we shall see in the next section).
- in the context of multiparty decision-making, the problem is made even more thorny because it also generates potential conflicts as to what should count as positive or negative arguments, when agents would maybe more easily come up with an agreement if two alternatives were compared.

To elaborate on the point mentioned above, we refer to a recent discussion on the UAI (Uncertainty in Artificial Intelligence) list where the problem of "where to draw the line" emerged as a thread of discussion. A very illustrative real example of this problem was given, as reported here:

> The contested 2000 US Presidential election and the question of "hanging chads." [...] in many instances the perforation was partial – leaving a hanging chad, a scrap of paper hanging from the voting card. So how was one to decide whether or not a partial perforation was or was not a vote for the position or person next to the perforation? One method, sometimes used, was to have the vote counter ask, "What intent does this perforation indicate?" Another approach was possible: It is useless or impossible to try to determine or guess the voter's intention. One

must instead ask whether this perforation looks more like a vote than a non-vote [2]."

So for instance two agents could discuss whether a given value should count as a positive goal or a negative argument, one arguing that this is not high enough a mark to be counted as a positive argument... while, it would be more practical to simply ask the agents to simply say whether they prefer an alternative versus another wrt. this given criterion.

3 Aggregation

Once you have decided what counts as arguments pro and con, for each possible decision, it is necessary to aggregate them to eventually decide what alternative to select. At this point, there is an important question to be asked: how is it that you handle potential interactions between arguments that refer to different criteria?

- if you assume that these interactions do not exist, or do not take them into account, then you might first aggregate arguments independently on each criterion, and then aggregate the resulting criteria using a given operator;
- if you take into account these interactions, then it is necessary to design an aggregation process that will aggregate arguments labelled by criteria.

There are many rational ways to aggregate sets of pro and cons. Bonnefon and Fargier [BF06] offer a nice overview of different possible approaches. These approaches take into account the fact that the arguments are bipolar and qualitative. The importance of arguments is described on totally ordered scale of magnitude. In order to compare these qualitative, bipolar sets, they present several procedures: the *Pareto comparison* (sets of arguments are compared as a problem of bi-criteria decision), the *implication rule* (this rule focuses on the most important arguments in the situation), the *cardinality rules* (based on a levelwise comparison by cardinality), and so on. The characterization of these rules was introduced in [DF05], and [BF06] present an extensive empirical assessment of the descriptive validity of these rules. What Amgoud *et al.* show in [ABP05] is that it is possible to retrieve various classical aggregation operators in their framework. They propose to compare decision in terms of positive and negative arguments (using a complex scheme for evaluating the strength of argument, which depends on three parameters : the *certainty level*, the *importance degree of the criterion*, and the *(dis)satisfaction degree of the criterion*). Two principles based on preference relation between the arguments are proposed : promotion focus (take into account only the supporting arguments) and preventing focus (considers only the arguments against decisions). They show that the presented framework captures different multiple criteria decision rules to select the best decision. The rule for the choice is characterized by the fact

[2] [P. Tiller, post on UAI list in response to L. Zadeh].

that the criteria have or not the same importance level. In this approach however, the potential interaction between arguments, as analysed in the seminal work of Dung [Dun95], is not considered. To the best of our knowledge, the *value-based argumentation* framework of [BC02] is the only approach so far that proposes to compute the acceptable arguments from a set of labelled arguments. Indeed, an argument refers to a given criterion (or "value" in the sense previously mentioned in the work of Atkinson [Atk05]). Argument systems, in the sense of Dung [Dun95], hence record interaction between arguments, possibly related to different values. *Audiences* are different ways to order those values. It is then possible to identify those arguments that will be accepted regardless of the chosen audience (*objectively* acceptable), while some others arguments can only be *subjectively* acceptable. In this case, the interaction between arguments pertaining to different criteria is fully recognized. The aggregation of values remains rather limited though, for it is only possible to order the values to reflect their degree of importance. So for instance it would not be possible to use an aggregation operator like the majority.

We conclude by a further remark related to the choice of the method of evaluation (intrinsic or pairwise). Both techniques may provide different, even contradictory, results, depending on the choice of the profile. Consider the following example. We assume that each criteria on an evaluation scale form 0 to 9. We take the neutral action to be $p = [5, 5, 5]$, meaning that the neutral point on each criteria is 5, and consider the following performance table :

	g_1	g_2	g_3
a	8	6	8
b	7	4	2
c	9	7	4

We will now use the following notation: $c \succ p$ *since* $[+, +, -]$, to specify that c is preferred to p because a strict majority of arguments (two arguments out of three) supports this proposition (only the last criteria disagrees with this). We will now compare the results obtained by aggregation, comparing the cases where an intrinsic or pairwise evaluation is used.

- in the case of intrinsic evaluation, we get $a \succ p$ *since* $[+, +, +]$ and $c \succ p$ *since* $[+, +, -]$. The obtained set of argument for a dominates that obtained for c: any rational aggregation method will give the outcome that $a \succ c$.
- in the case of a pairwise comparison (we don't need to use the neutral action then), we have on the other hand $c \succ a$ *since* $[+, +, -]$, hence $c \succ a$

This simply illustrates that both methods may return contradictory results, which is easily explained by the fact that the categorisation as argument pro or con may make the preference model rather coarse-grained (of course, we do not discuss here the possibility of using a more detailed bipolar scale, as mentionned earlier in this paper).

4 Conclusion

The primary aim of this paper was to offer a critical review of existing approaches adopting an argumentative stance towards decision-making, adopting the viewpoint of (multicriteria) decision theory. We emphasized in particular that arguments pro or against a given action are generally regarded as resulting from a comparison against a neutral action (drawn from the agent's preferences). This intrinsic evaluation technique departs from the pairwise evaluation, and raises some difficulties it is good to be aware of. On the other hand, it appears that these approaches seem to have much to offer to decision models, because they allow a great expressivity in the specification of agents' (possibly partial) preferences, and because they make explicit many aspects that are usually somewhat hidden in decision models. At the level of aggregation, despite recent progresses, the question of how the multicriteria feature should be taken into account when it comes to aggregating several arguments (that may potentially interact and refer to different criteria) remains largely unexplored.

Acknowledgments. We would like to thank the reviewers for their comments, as well as the audience of the First Workshop on Logics and Collective Decision-making (2007) in Lille, where an earlier version of this work was presented.

References

[ABCM06] Atkinson, K., Bench-Capon, T.J.M., Modgil, S.: Argumentation for decision support. In: Proc. of the 17th International Conf. on Database and Expert Systems Applications, pp. 822–831 (2006)

[ABP05] Amgoud, L., Bonnefon, J.-F., Prade, H.: An Argumentation-based Approach to Multiple Criteria Decision. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 269–280. Springer, Heidelberg (2005)

[aC96] Bana a Costa, C.: Les problématiques de l'aide à la décision: vers l'enrichissement de la trilogie choix-tri-rangement. Recherche Operationnelle 30(2), 191–216 (1996)

[AP06] Amgoud, L., Prade, H.: Explaining qualitative decision under uncertainty by argumentation. In: Proc. of the 21st AAAI Conference on Artificial Intelligence (2006)

[Atk05] Atkinson, K.: What Should We Do?: Computational Representation of Persuasive argument in practical reasoning. PhD thesis (2005)

[Atk06] Atkinson, K.: Value-based argumentation for democratic support. In: Proc. of the 1st International Conf. on Computational Models of Natural Argument, pp. 47–58. IOS Press, Amsterdam (2006)

[BC02] Bench-Capon, T.J.M.: Value-based argumentation frameworks. In: Proc. of 9th International Workshop on Non Monotonic Reasoning, pp. 443–454 (2002)

[BCP06] Bench-Capon, T.J.M., Prakken, H.: Justifying Actions by Accruing Arguments. In: Proc. of the 1st International Conf. on Computational Models of Argument, pp. 247–258. IOS Press, Amsterdam (2006)

[BF06] Bonnefon, J.-F, Fargier, H.: Comparing sets of positive and negative arguments: Empirical assessment of seven qualitative rules. In: Proc. of 17th European Conf. on Artificial Intelligence, IOS Press, Amsterdam (2006)

[BG96] Bonet, B., Geffner, H.: Arguing for Decisions: A Qualitative Model of Decision Making. In: Proc. of the 12th Conf. on Uncertainty in Artificial Intelligence, pp. 98–105 (1996)

[BMP+00] Bouyssou, D., Marchant, T., Pirlot, M., Perny, P., Tsoukiàs, A., Vincke, P.: Evaluation and decision models: a critical perspective. Kluwer Academic Publishers, Dordrecht (2000)

[BS02] Belton, V., Stewart, T.: Muliple Criteria Decision Analysis: an Integrated Approach. Kluwer Academic Publishers, Dordrecht (2002)

[CLS05] Cayrol, C., Lagasquie-Schiex, M.-C.: Graduality in argumentation. Journal of Artificial Intelligence Research 23, 245–297 (2005)

[DF05] Dubois, D., Fargier, H.: On the qualitative comparison of sets of positive and negative affects. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 305–316. Springer, Heidelberg (2005)

[Dun95] Dung, P.M.: On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-person games. Artificial Intelligence 77(2), 321–358 (1995)

[FBB80] Fox, J., Barber, D., Bardhan, K.D.: Alternatives to bayes? A quantitative comparison with rule-based diagnostic inference. Methods of Information in Medicine 19(4), 210–215 (1980)

[FP97] Fox, J., Parsons, S.: On Using Arguments for Reasoning about Actions and Values. In: Proc. of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning, pp. 55–63. AAAI Press, Stanford (1997)

[FP98] Fox, J., Parsons, S.: Arguing about beliefs and actions. In: Applications of Uncertainty Formalisms, pp. 266–302 (1998)

[KP98] Karacapilidis, N.I., Papadias, D.: Hermes: Supporting argumentative discourse in multi-agent decision making. In: AAAI/IAAI, pp. 827–832 (1998)

[MM07] Morge, M., Mancarella, P.: The hedgehog and the fox. an argumentation-based decision support system. In: Proc. of the 4th International Workshop on Argumentation in Multi-Agent Systems (2007)

[Mod06] Modgil, S.: Value based argumentation in hierarchical argumentation. In: Proc. of the 1st International Conf. on Computational Models of Natural Argument (2006)

[PJ98] Parsons, S., Jennings, N.R.: Argumentation and multi-agent decision making. In: Proc. of the AAAI Spring Symposium on Interactive and Mixed-Initiative Decision Making, pp. 89–91 (1998)

[Pol87] Pollock, J.L.: Defeasible reasoning. Cognitive Science 11, 481–518 (1987)

[PV02] Prakken, H., Vreeswijk, G.: Logical systems for defeasible argumentation. In: Handbook of Philosophical Logic, vol. 4, pp. 219–318. Kluwer Academic Publishers, Dordrecht (2002)

[Tso07] Tsoukiàs, A.: On the concept of decision aiding process. Annals of Operations Research, appeared previously as DIMACS 2003-38 technical report, Rutgers University (to appear)

[Wal96] Walton, D.N.: Argumentation schemes for Presumptive Reasoning. Erlbaum, Mahwah (1996)

Shared Ordered Binary Decision Diagrams for Dempster-Shafer Theory

Norbert Lehmann

National ICT Australia Ltd
University of New South Wales, Australia
norbert.lehmann@unifr.ch

Abstract. The binary representation is widely used for representing focal sets of Dempster-Shafer belief functions because it allows to compute efficiently all relevant operations. However, as its space requirement grows exponentially with the number of variables involved, computations may become prohibitive or even impossible for belief functions with larger domains. This paper proposes shared ordered binary decision diagrams for representing focal sets. This not only allows to compute efficiently all relevant operations, but also turns out to be a compact representation of focal sets.

1 Introduction

Dempster-Shafer theory is a well founded and widely accepted theory for uncertain reasoning and includes probability theory as a special case. Its main building blocks are *belief potentials* together with two operations called *combination* and *marginalization*. Given a set of variables V, a belief potential φ on domain $d(\varphi) \subseteq V$ typically encodes a piece of evidence or an uncertain event.

The main computational problem in Dempster-Shafer theory is to marginalize the joint combination of belief potentials $\varphi_1, \ldots, \varphi_n$ to a certain domain of interest $Q \subseteq V$, i.e. to compute $(\varphi_1 \otimes \cdots \otimes \varphi_n)^{\downarrow Q}$. However, it is often computationally prohibitive or even impossible to compute first the joint combination. Instead, the variables in $V \setminus Q$ are eliminated one after another according to a variable elimination sequence. At each step of this elimination process, first the joint combination of a subset of the remaining potentials is computed and then the resulting potential is marginalized to a smaller domain where the current elimination variable does not occur anymore.

For the elimination process to be executed as fast as possible, belief potentials have to be encoded appropriately. Above all, the encoding of the focal sets of belief potentials is particularly important and should allow to execute efficiently the following four operations: (1) *intersection*, (2) *projection*, (3) *extension* and (4) *equality testing*.

In the last few years, research has focussed on the binary representation of focal sets. This encoding has a number of advantages, for example the above four operations can easily be implemented and are executed efficiently on current microprocessors. However, as its space requirement grows linearly with the

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 320–331, 2007.

number of focal sets and exponentially with the number of variables involved, computations may become prohibitive or even impossible for belief potentials with large domains.

In the knowledge compilation map [1,2] of Darwiche and Marquis, the binary representation of focal sets corresponds to the language MODS. Interestingly, the knowledge compilation map contains besides MODS another language which supports the above four operations in polytime. This language is called $\text{OBDD}_<$ and corresponds to the language of *ordered binary decision diagrams*. Although the size of an OBDD can be exponential in the number of variables, this upper limit is only rarely attaint in practice. Therefore, the language of OBDD is an interesting candidate for encoding the focal sets of belief potentials. A possible solution would be to encode the focal sets as a sequence of OBDDs. However, this paper goes a step further and investigates on the usability of *shared ordered binary decision diagrams* for encoding the focal sets of belief potentials. This encoding not only allows to compute efficiently all four relevant operations for focal sets, but often also turns out to be a compact representation of focal sets.

This paper is structured as follows. We start in Section 2 with an overview of Dempster-Shafer theory. Then, Section 3 introduces the binary representation of focal sets. Section 4 gives a short introduction to BDDs and Section 5 then shows how to encode the focal sets of belief potentials using SOBDDs. Finally, we close with some concluding remarks in Section 6.

2 Multivariate Dempster-Shafer Theory

The foundation of Dempster-Shafer theory [3,4,5,6] was laid in [7,8] where Dempster studied upper and lower bounds of probability distributions induced by a multivalued mapping. Shafer then continued Dempster's work and developed a theory of evidence [9]. He proposed to call set functions having the structure of Dempster's lower probabilities *belief functions*. Nowadays, Dempster-Shafer theory is often used to represent uncertain knowledge. Its building blocks are *belief potentials* which encode pieces of evidence.

In this paper we consider multivariate belief potentials only. Let V denote the set of all variables. Every variable $x \in V$ has a finite set Θ_x of possible values. In order to make life easier, we restrict ourself to binary variables[1]. Without loss of generality, we can then assume that $\Theta_x = \{0, 1\}$ for all $x \in V$. Throughout this paper, if not specified otherwise, let $D = \{x_1, \ldots, x_n\}$ be a subset of variables. The elements of the corresponding cartesian product $\Theta_D = \Theta_{x_1} \times \cdots \times \Theta_{x_n}$ are called *configurations* of D. Note that Θ_D has exactly 2^n configurations.

2.1 Different Representations

Similar to complex numbers where $c \in \mathbb{C}$ can be represented in polar or rectangular form, there are different ways to represent a belief potential φ. It can

[1] The more general case of non-binary variables leads to shared ordered multi-valued decision diagrams (SOMDD) [10].

be represented as a *mass function* $[\varphi]_m$, as a *belief function* $[\varphi]_b$ or as a *commonality function* $[\varphi]_q$. The mass function representation has certain advantages compared to the other representations. Above all, the main operations for belief potentials can be stated easier using mass functions.

Mass Functions. A *mass function* $[\varphi]_m$ on D assigns to every subset X of Θ_D a value in $[0,1]$, that is $[\varphi]_m : 2^{\Theta_D} \rightarrow [0,1]$. The following condition must be satisfied:

$$\sum_{X \subseteq \Theta_D} [\varphi(X)]_m = 1. \tag{1}$$

Sometimes, a second condition, $[\varphi(\emptyset)]_m = 0$, is imposed. A mass function for which this additional condition holds is called *normalized*, otherwise it is called *unnormalized*. A set $X \subseteq \Theta_D$ for which $[\varphi(X)]_m > 0$ is called *focal set* of φ. Note that a belief potential φ is completely specified by the collection $\{(X_1, [\varphi(X_1)]_m), \ldots (X_\ell, [\varphi(X_\ell)]_m)\}$ where $X_1, \ldots, X_\ell$ are the focal sets of φ. The set of all focal sets of a belief potential φ is denoted by $FS(\varphi)$.

2.2 Operations for Belief Potentials

The two main operations for belief potentials are *combination* and *marginalization*. Intuitively, these two operations correspond to aggregation and focusing. Suppose φ_1 and φ_2 are potentials on D_1 and D_2. The *combination* of these two potentials is given by Equation (2) and produces an unnormalized potential on domain $D = D_1 \cup D_2$. Similarly, suppose that φ is a potential on D and $C \subseteq D$. The *marginalization* of φ to C is given by Equation (3) and produces a potential on domain C.

$$[\varphi_1 \otimes \varphi_2(X)]_m = \sum_{X_1^{\uparrow D} \cap X_2^{\uparrow D} = X} [\varphi_1(X_1)]_m \cdot [\varphi_2(X_2)]_m \tag{2}$$

$$[\varphi^{\downarrow C}(X)]_m = \sum_{Y^{\downarrow C} = X} [\varphi(Y)]_m \tag{3}$$

Another useful operation is *extension* and is given by Equation 4. It takes a potential φ on domain $C \subseteq D$ and produces a potential on domain D. Using the *extension* operation, combination can be rewritten as given in Equation (5).

$$[\varphi^{\uparrow D}(X)]_m = \begin{cases} [\varphi(Y)]_m & \text{if } X = Y^{\uparrow D}, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

$$[\varphi_1 \otimes \varphi_2(X)]_m = \sum_{X_1 \cap X_2 = X} [\varphi_1^{\uparrow D}(X_1)]_m \cdot [\varphi_2^{\uparrow D}(X_2)]_m \tag{5}$$

2.3 Operations for Focal Sets

Combination and *marginalization* use some operations for focal sets. These operations are:

– *intersection* $(X_1 \cap X_2)$,
– *projection* $(X^{\downarrow C})$,
– *extension* $(X^{\uparrow D})$ and
– *equality testing* $(X = Y)$.

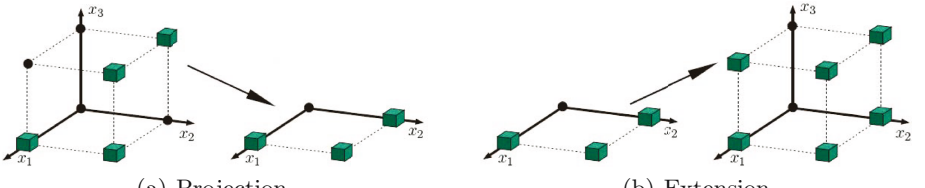
Equality testing is used to group together identical focal sets when combining or marginalizing belief potentials.

Projection of Focal Sets. It is computed when a belief potential is marginalized. If $\mathbf{c} = (r_1, \ldots, r_n) \subseteq \Theta_D$ is a configuration of D and $C \subseteq D$, then $\mathbf{c}^{\downarrow C}$ denotes the projection of $\mathbf{c}$ to C. It is obtained by removing all components of $\mathbf{c}$ which correspond to variables in $D \setminus C$. For a focal set $X \subseteq \Theta_D$, the projection of X to C is a subset of Θ_C and is denoted by $X^{\downarrow C}$. It is obtained by projecting each element of X to C, that is $X^{\downarrow C} = \{\mathbf{c}^{\downarrow C} : \mathbf{c} \in X\}$.

Extension of Focal Sets. It is computed when two belief potentials are combined. For a focal set $X \subseteq \Theta_C$ and $C \subseteq D$, the extension of X to D is a subset of Θ_D and is denoted by $X^{\uparrow D}$. It is given by the cylindrical extension $X^{\uparrow D} = X \times \Theta_{D \setminus C}$.

Example 1. Let $D = \{x_1, x_2, x_3\}$. Θ_D consists of 8 configurations and is given by $\Theta_D = \{(0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0), (1,1,1)\}$. There are 256 different subsets $X \subseteq \Theta_D$. On the very left side of Figure 1, the subset $X = \{(0,1,1), (1,0,0), (1,1,0), (1,1,1)\}$ is represented in a 3-dimensional cube of which the axes are given by the variables x_1, x_2 and x_3. The projection of X to the set of variables $C = \{x_1, x_2\}$ is given by $Y = X^{\downarrow C} = \{(0,1), (1,0), (1,1)\}$ and is also represented on the left side of Figure 1.

Example 2. Let $C = \{x_1, x_2\}$. Θ_C then consists of 4 configurations. The subset $Y = \{(0,1), (1,0), (1,1)\}$ is represented on the right side of Figure 1. For $D = \{x_1, x_2, x_3\}$, the extension of Y to D corresponds to the cylindrical extension given by $Z = Y^{\uparrow D} = \{(0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0), (1,1,1)\}$ and is represented on the very right side of Figure 1.



(a) Projection (b) Extension

Fig. 1. Projection and extension of a focal set

3 The Binary Representation

A representation which was already proposed in [11] is to use *bitstrings* for focal sets. This representation is based on a global ordering $<_g$ of all variables called the *variable encoding order*. We suppose in the following that all given sets of variables respect this global ordering, meaning that $x_i, x_k \in D$ and $i < k$ implies $x_i <_g x_k$.

In [12], different algorithms were presented which efficiently perform the operations on belief potentials where focal sets are encoded using the binary representation. These algorithms rely on a small set of basic functions for manipulating arbitrary long bitstrings and can be implemented very easily in programming languages which support arbitrary long integer numbers. Belief function computations then turn out to be extremely efficient if the domains of the belief potentials are relatively small.

3.1 Focal Sets as Bitstrings

Let $D = \{x_1, \ldots, x_n\} \subseteq V$ be a subset of variables respecting the variable encoding order. $\Theta_D = \{\mathbf{c}_0, \ldots, \mathbf{c}_{m-1}\}$ consists of exactly $m = 2^n$ configurations. The index r of each configuration $\mathbf{c}_r = (r_1, \ldots, r_n) \in \Theta_D$ is determined by the values r_1 to r_n and

$$r = \sum_{i=1}^{n} \left(r_i \cdot 2^{n-i}\right). \tag{6}$$

A focal set $X \subseteq \Theta_D$ can now be represented unambiguously by a bitstring $\mathcal{B}_D(X) = \langle b_{m-1} \cdots b_0 \rangle$ where

$$b_i = \begin{cases} 1 & \text{if } \mathbf{c}_i \in X, \\ 0 & \text{otherwise.} \end{cases}$$

The bitstring $\mathcal{B}_D(X)$ determines a unique integer number which is composed of exactly 2^n bits.

Example 3. Let $D = \{x_1, x_2, x_3\}$, $C = \{x_1, x_2\}$ and $X \subseteq \Theta_D$ given by $X = \{(0,1,1), (1,0,0), (1,1,0), (1,1,1)\}$. Figure 1 displays from left to right the focal sets X, $Y = X^{\downarrow C}$ and $Z = Y^{\uparrow D}$. Therefore, $Y \subseteq \Theta_C$ and $Z \subseteq \Theta_D$. The following bitstrings are associated with these three focal sets:

$$\mathcal{B}_D(X) = \langle 01101100 \rangle, \mathcal{B}_C(Y) = \langle 1110 \rangle \text{ and } \mathcal{B}_D(Z) = \langle 11111100 \rangle.$$

3.2 Operations for Focal Sets

Several algorithms were presented in [12] to compute the relevant operations on focal sets. For $D = \{x_1, \ldots, x_n\}$and $C \subseteq D$, two of these algorithms compute the *projection* $\mathcal{B}_C(X^{\downarrow C})$ and the *extension* $\mathcal{B}_D(Y^{\uparrow D})$ for focal sets $X \subseteq \Theta_D$ and $Y \subseteq \Theta_C$ given as bitstrings $\mathcal{B}_D(X)$ and $\mathcal{B}_D(Y)$, respectively.

In the following, we describe only the special case where $C = D \setminus \{x_i\}$ for an arbitrary variable $x_i \in D$. We introduce first a notation which is helpful for describing projection and extension of focal sets given as bitstrings. For this purpose, the set of variables D is divided into

$$D = \{\underbrace{x_1, \ldots, x_{i-1}}_{U_D(x_i)}, x_i, \underbrace{x_{i+1}, \ldots, x_n}_{V_D(x_i)}\}.$$

Thus, $U_D(x_i)$ corresponds to the set of variables in D which are to the left of x_i and similarly $V_D(x_i)$ to the set of variables in D which are to the right of x_i. In the following, let $u = |\Theta_{U_D(x_i)}| = 2^{i-1}$ and $v = |\Theta_{V_D(x_i)}| = 2^{n-i}$.

Projection of Focal Sets. Let $\mathcal{B}_D(X)$ be the bitstring for a focal set $X \subseteq \Theta_D$. It has 2^n bits and is composed of $2u$ smaller blocks each having v bits:

$$\mathcal{B}_D(X) = \langle \mathcal{B}_2^u \mathcal{B}_1^u \mathcal{B}_2^{u-1} \mathcal{B}_1^{u-1} \cdots \mathcal{B}_2^2 \mathcal{B}_1^2 \mathcal{B}_2^1 \mathcal{B}_1^1 \rangle$$

The corresponding bitstring $\mathcal{B}_C(X^{\downarrow C})$ has 2^{n-1} bits and is composed of u blocks:

$$\mathcal{B}_C(X^{\downarrow C}) = \langle \mathcal{R}^u \mathcal{R}^{u-1} \cdots \mathcal{R}^2 \mathcal{R}^1 \rangle$$

Each one of these smaller blocks $\mathcal{R}^k$ has v bits and is equal to the logical inclusive OR of the blocks $\mathcal{B}_2^k$ and $\mathcal{B}_1^k$.

Extension of Focal Sets. Let $\mathcal{B}_C(Y)$ be the bitstring for a focal set $Y \subseteq \Theta_C$. It has 2^{n-1} bits and is composed of u smaller blocks each having v bits:

$$\mathcal{B}_C(Y) = \langle \mathcal{B}^u \mathcal{B}^{u-1} \cdots \mathcal{B}^2 \mathcal{B}^1 \rangle$$

The corresponding bitstring $\mathcal{B}_D(Y^{\uparrow D})$ has 2^n bits and is composed of $2u$ blocks:

$$\mathcal{B}_D(Y^{\uparrow D}) = \langle \mathcal{B}^u \mathcal{B}^u \mathcal{B}^{u-1} \mathcal{B}^{u-1} \cdots \mathcal{B}^2 \mathcal{B}^2 \mathcal{B}^1 \mathcal{B}^1 \rangle$$

Therefore, each one of the smaller blocks is repeated twice.

3.3 The Main Problem: Space Requirement

The space requirement of the binary representation of focal sets grows linearly with the number of focal sets and exponentially with the number of variables involved. Therefore, belief function computations may become prohibitive or even impossible for belief potentials with large domains.

4 Binary Decision Diagrams

A Binary Decision Diagram (BDD) [13,14,15] is a data structure which is used to represent and manipulate Boolean functions. It corresponds to a rooted directed acyclic graph with one or two terminal nodes and several decision nodes. The terminal nodes are called *0-sink* and *1-sink* and represent the Boolean constants $\bot$ and $\top$, respectively. Every decision node u has a label $var(u)$ and two child nodes called $high(u)$ and $low(u)$, respectively.

Example 4. The BDD shown on the right side of Figure 2 consists of three decision nodes and two terminal nodes. Typically, $high(u)$ of a decision node u is drawn with a solid flash whereas $low(u)$ is drawn with a dashed flash.

4.1 Boolean Functions

Boolean functions are functions from $\mathbb{B}^n \to \mathbb{B}$ where $\mathbb{B} = \{0, 1\}$. Its building blocks are Boolean variables $x_1, \ldots, x_n$ and the Boolean constants $\bot$ and $\top$. If f and g are Boolean functions, then also $\neg f$, $f \wedge g$ and $f \vee g$.

Definition 1. *Let $f(x_1, \ldots, x_n)$ be a Boolean function. Then*

$$f_{x_i \leftarrow 0} = f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \tag{7}$$

$$f_{x_i \leftarrow 1} = f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) \tag{8}$$

are the positive and negative cofactors of f with respect to x_i.

4.2 Construction of BDDs

The construction of a BDD for a Boolean function f is based on the Shannon expansion of f. This states that

$$f(x_1, \ldots, x_n) \equiv (x \wedge f_{x \leftarrow 1}) \vee (\neg x \wedge f_{x \leftarrow 0}) \tag{9}$$

for all Boolean functions $f(x_1, \ldots, x_n)$ and variables $x \in \{x_1, \ldots, x_n\}$. It allows to construct a BDD for a given Boolean function $f(x_1, \ldots, x_n)$ by the recursive procedure shown by Algorithm 1.

Algorithm 1. BUILD($f(x_1, \ldots, x_n)$)

input : Boolean function $f(x_1, \ldots, x_n)$
output: BDD node which represents f

if ($f \equiv \bot$) **then return** *0-sink*;
if ($f \equiv \top$) **then return** *1-sink*;
Select variable x;
  t = BUILD($f_{x \leftarrow 1}$);
  e = BUILD($f_{x \leftarrow 0}$);
  return *MK(x, t, e)*;

4.3 Reduction Rules

In order to minimize the size of BDDs, some reduction rules are always applied:

1. Two terminal nodes with the same label are merged.
2. Two decision nodes with the same label and the same children are merged.
3. If a decision node has identical children, it is removed from the graph and all incoming edges are redirected to its child.

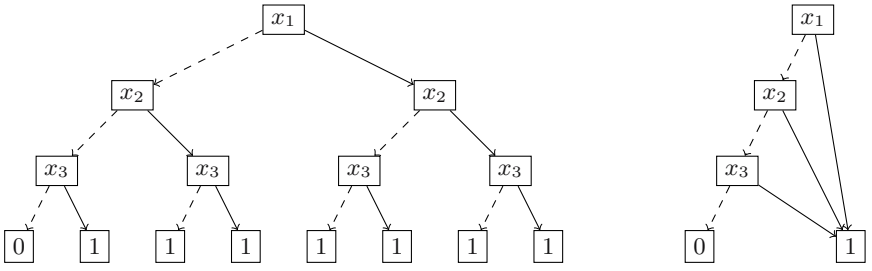


Fig. 2. Reduction rules for Boolean function $f(x_1, x_2, x_3) = x_1 \vee x_2 \vee x_3$

Example 5. A BDD for the Boolean function $f(x_1, x_2, x_3) = x_1 \vee x_2 \vee x_3$ is shown on the left side of Figure 2. After the application of the above three reduction rules, the OBDD shown on the right side of Figure 2 is obtained.

In Algorithm 1, the function MK($var, high, low$) takes care of these reduction rules which may considerably decrease the number of nodes of a BDD. In the following, we restrict the discussion to reduced BDDs and refer to them simply as BDDs.

4.4 Ordering

A heavy impact on the size of BDDs has the order in which the variables are selected when Shannon expansion is applied. If the variables are selected according to a predefined total order, then the variables on all paths from the root node of a BDD occur in the same ordering. This class of BDDs is called *ordered binary decision diagrams* (OBDD).

Every node u of an OBDD represents a Boolean function f^u. An important property of OBDDs is that they are a canonical representation of Boolean functions. This means that equivalent Boolean functions are represented by the same OBDD node.

5 Binary Decision Diagrams for Dempster-Shafer Theory

An OBDD u represents a corresponding Boolean function f^u. Consequently, a collection $\{f^{u_1}, \ldots, f^{u_\ell}\}$ of Boolean functions can be represented by a collection $\{u_1, \ldots, u_\ell\}$ of OBDDs. However, these OBDDs might share common factors. Therefore, it is worthwhile to extend sharing to different OBDDs so that subgraphs can be used by several OBDDs. The class of diagrams of this type is called *shared ordered binary decision diagrams* (SOBDD) [16,17] and is given by a forest of *multi-rooted* directed acyclic graphs. Note that each node of an SOBDD together with the set of all its successor nodes corresponds to an OBDD.

5.1 Focal Sets and Shared Ordered Binary Decision Diagrams

A belief potential φ on domain $D = \{x_1, \ldots, x_n\}$ is completely specified by the collection $\{(X_1, [\varphi(X_1)]_m), \ldots, (X_\ell, [\varphi(X_\ell)]_m)\}$ where $X_1, \ldots, X_\ell \in FS(\varphi)$ are the focal sets of φ. We are now going to associate a SOBDD to these focal sets. For that purpose, first, we have to associate a Boolean function $f_X(x_1, \ldots, x_n)$ to every focal set $X \in FS(\varphi)$. This can be done by

$$f_X(\mathbf{r}_1, \ldots, \mathbf{r}_n) = 1 \quad \Longleftrightarrow \quad (\mathbf{r}_1, \ldots, \mathbf{r}_n) \in X. \tag{10}$$

Therefore, every focal set $X \subseteq \Theta_D$ has an associated Boolean function f_X. Given a *variable encoding order*, f_X is unequivocally determined by an OBDD u_X. The collection of OBDDs $\{u_{X_1}, \ldots, u_{X_\ell}\}$ then forms a SOBDD which contains for every focal set $X \in FS(\varphi)$ a node which represents the associated Boolean function f_X.

Example 6. Let $D = \{x_1, x_2\}$ and φ be a belief potential with focal sets $X_1 = \{(1,1)\}$, $X_2 = \{(0,1),(1,1)\}$, $X_3 = \{(0,1),(1,0),(1,1)\}$, $X_4 = \Theta_D$ and corresponding masses 0.1, 0.2, 0.3 and 0.4, respectively. These four focal sets correspond to the Boolean functions $f_{X_1}(x_1, x_2) = x_1 \wedge x_2$, $f_{X_2}(x_1, x_2) = x_2$, $f_{X_3}(x_1, x_2) = x_1 \vee x_2$ and $f_{X_4}(x_1, x_2) = \top$, respectively. If $x_1 < x_2$ is the variable ordering then the collection $\{f_{X_1}, f_{X_2}, f_{X_3}, f_{X_4}\}$ of Boolean functions is represented by the SOBDD shown in Figure 3. The four corresponding nodes are indicated in the graph.

It is well known that there are classes of Boolean functions that have OBDDs of size exponential in the number of variables, regardless of the ordering chosen [18]. However, this upper limit is rarely obtained in practice. In most cases, a collection of boolean functions can be represented very compactly using an SOBBD.

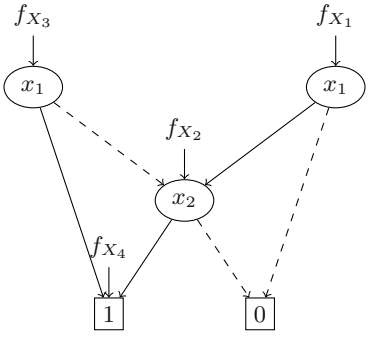


Fig. 3. SOBDD for the focal sets of the belief potential φ

5.2 Operations on Focal Sets

The four main operations on focal sets translate into corresponding operations on OBDDs.

Intersection. The intersection of two OBDDs u and v is performed by Algorithm 2. An interesting point to note is that a *cache* is used to store computed results. Without the cache, the runtime of AND would grow exponentially with the number of variables. If all computed results are stored then the time complexity of intersection is polynomial in the product of the size of each OBDD. However, as there may be a huge number of intermediate results it is often not possible to store all intermediate results. As a consequence, the worst case performance of practical implementations of AND is exponential in the number of variables, but the exponential behavior is rarely observed.

Algorithm 2. AND(u, v)

input : OBDD nodes u and v representing boolean functions f^u and f^v
output: OBDD node representing the boolean function $f^u \wedge f^v$

if (*terminal case*) **then return** *terminal result*;
if (*cache has entry* $\{u, v\}$) **then return** *cache result*;
With x being the top variable of $\{u, v\}$
  $t = \text{AND}(f^u_{x\leftarrow 1}, f^v_{x\leftarrow 1})$;
  $e = \text{AND}(f^u_{x\leftarrow 0}, f^v_{x\leftarrow 0})$;
  $r = \text{MK}(x, t, e)$;
  InsertIntoCache$(\{u, v\}, r)$;
  return r;

Projection. Let $D = \{x_1, \ldots, x_n\}$ and $C = D \setminus \{x\}$ for $x \in D$. In addition, let $X \subseteq \Theta_D$ be a focal set with associated Boolean function $f(x_1, \ldots, x_n)$. Using OBDDs, the projection of X to C can then be obtained by *existential quantification* denoted by $\exists x.f(x_1, \ldots, x_n)$. It is given as follows:

$$\exists x.f(x_1, \ldots, x_n) = f_{x\leftarrow 0} \vee f_{x\leftarrow 1}$$

Therefore, projection can be implemented by two calls to RESTRICT and one call to OR. These are both functions which manipulate OBDDs. For example, the function OR is very similar to Algorithm 2. Concerning the complexity of this operation, the same remarks as for intersection apply.

Extension. If focal sets are represented using an SOBDD, then this operation is not needed. Let $C \subseteq D$, $X \subseteq \Theta_C$ and $Y = X^{\uparrow D}$. The corresponding Boolean functions f_X and f_Y are equivalent and consequently, the *same* OBDD node is associated with f_X and f_Y.

Equality Testing. This can be performed extremely efficient using OBDDS. Because OBDDs are a canonical representation of Boolean functions, the time

complexity of equality testing is $O(1)$. Two Boolean functions are equivalent if and only if they are represented by the same OBDD, i.e. if the same OBBD node is associated with both Boolean functions.

6 Conclusion and Outlook

The binary representation of focal sets is extremely efficient if the domains of the belief potentials involved is small. However, as its space requirement grows linearly with the number of focal sets and exponentially with the number of variables involved, belief function computations may become prohibitive or even impossible for belief potentials with large domains.

This paper has taken a look on the use of *shared ordered binary decision diagrams* for encoding the focal sets of belief functions. This encoding allows to execute efficiently the four main operations on focal sets. But most importantly, it often also turns out to be a very compact representation of focal sets.

Practical tests using examples from different application domains have now to be conducted in order to compare these two representations. Preliminary tests based on the freely available software package CUDD [19] are very promising. Our first conclusions obtained from these tests indicate that the binary representation is more suited for belief function computations involving small domains. If larger domains are involved, then the SOBDD representation is much better suited. As a consequence, the best encoding for focal sets is most probably based on a hybrid approach.

Acknowledgements

Research supported by Grant PA002–109013 of the Swiss National Science Foundation.

References

1. Darwiche, A., Marquis, P.: A knowledge compilation map. J. Artif. Intell. Res (JAIR) 17, 229–264 (2002)
2. Wachter, M., Haenni, R.: Propositional DAGs: a new graph-based language for representing Boolean functions. In: Doherty, P., Mylopoulos, J., Welty, C. (eds.) KR'06, 10th International Conference on Principles of Knowledge Representation and Reasoning, Lake District, U.K., pp. 277–285. AAAI Press, Stanford (2006)
3. Kohlas, J., Monney, P.: A Mathematical Theory of Hints. An Approach to the Dempster-Shafer Theory of Evidence. Lecture Notes in Economics and Mathematical Systems, vol. 425. Springer, Heidelberg (1995)
4. Kong, A.: Multivariate Belief Functions and Graphical Models. PhD thesis, Department of Statistics, Harvard University (1986)
5. Smets, P.: Belief functions. In: Smets, P., Mamdani, A., Dubois, D., Prade, H. (eds.) Non-Standard Logics for Automated Reasoning, pp. 253–286. Academic Press, London (1988)

6. Thoma, H.M.: Belief function computations. In: Goodman, I.R., Gupta, M.M., Nguyen, H.T., Rogers, G.S. (eds.) Conditional Logic in Expert Systems, pp. 269–307. Elsevier, Amsterdam (1991)
7. Dempster, A.P.: Upper and lower probabilities induced by a multivalued mapping. Annals of Mathematical Statistics 38, 325–339 (1967)
8. Dempster, A.P.: A generalization of Bayesian inference. Journal of the Royal Statistical Society 30 (Series B), 205–247 (1968)
9. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
10. Wachter, M., Haenni, R.: Multi-state directed acyclic graphs. In: Kobti, Z., Wu, D. (eds.) CanAI'07, 20th Canadian Conference on Artificial Intelligence. LNCS (LNAI), vol. 4509, pp. 464–475, Montréal, Canada (2007)
11. Xu, H., Kennes, R.: Steps toward efficient implementation of dempster-shafer theory. In: Yager, R.R., Kacprzyk, J., Fedrizzi, M. (eds.) Advances in the Dempster-Shafer Theory of Evidence, pp. 153–174. John Wiley and Sons, New York (1994)
12. Haenni, R., Lehmann, N.: Implementing belief function computations. International Journal of Intelligent Systems 18(1), 31–49 (2003)
13. Bryant, R.E.: Graph-based algorithms for Boolean function manipulation. IEEE Transactions on Computers C-35(8), 677–691 (1986)
14. Bryant, R.E.: Symbolic Boolean manipulation with ordered binary-decision diagrams. ACM Computing Surveys 24(3), 293–318 (1992)
15. Somenzi, F.: Binary decision diagrams (1999)
16. Minato, S.I., Ishiura, N., Yajima, S.: Shared binary decision diagram with attributed edges for efficient boolean function manipulation. In: DAC '90: Proceedings of the 27th ACM/IEEE conference on Design automation, pp. 52–57. ACM Press, New York, NY, USA (1990)
17. Brace, K.S., Rudell, R.L., Bryant, R.E.: Efficient implementation of a bdd package. In: DAC, pp. 40–45 (1990)
18. Bryant, R.E.: On the complexity of VLSI implementations and graph representations of Boolean functions with application to integer multiplication. IEEE Transactions on Computers 40(2), 205–213 (1991)
19. Somenzi, F.: CUDD: CU decision diagram package release (1998)

Cautious Conjunctive Merging of Belief Functions

Sebastien Destercke[1,2], Didier Dubois[2], and Eric Chojnacki[1]

[1] Institut de Radioprotection et de Sûreté Nucléaire, DPAM, SEMIC, LIMSI, Cadarache, France
{sebastien.destercke,eric.chojnacki}@irsn.fr
[2] Toulouse Institute of Research in Computer Science (IRIT), Toulouse, France
{Destercke,Dubois}@irit.fr

Abstract. When merging belief functions, Dempster rule of combination is justified only when information sources can be considered as independent. When this is not the case, one must find out a cautious merging rule that adds a minimal amount of information to the inputs. Such a rule is said to follow the principle of minimal commitment. Some conditions it should comply with are studied. A cautious merging rule based on maximizing expected cardinality of the resulting belief function is proposed. It recovers the minimum operation when specialized to possibility distributions. This form of the minimal commitment principle is discussed, in particular its discriminating power and its justification when some conflict is present between the belief functions.

Keywords: belief functions, least commitment, dependence.

1 Introduction

There exist many fusion rules in the theory of belief functions [13]. When several sources deliver information over a common frame of discernment, combining belief functions by Dempster's rule [4] is justified only when the sources can be assumed to be independent. When such an assumption is unrealistic and when the precise dependence structure between sources cannot be known, an alternative is to adopt a conservative approach to the merging of the belief functions (i.e. by adding no extra information nor assumption in the combination process). Adopting such a cautious attitude means that we apply the "least commitment principle", which states that one should never presuppose more beliefs than justified. This principle is basic in the frameworks of possibility theory, imprecise probability [15], and the Transferable Belief Model (TBM) [14]. It can be naturally exploited for cautious merging belief functions.

In this paper, we study general properties that a merging rule satisfying the least commitment principle should follow when the sources are logically consistent with one another. An idempotent cautious merging rule generalizing the minimum rule of possibility theory is proposed. Section 2 recalls some basics about belief functions. Section 3 recalls an approach to the conjunctive merging

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 332–343, 2007.

of belief functions proposed by Dubois and Yager in the early nineties and shows it provides a natural least committed idempotent merging rule for belief functions, where least commitment comes down to maximizing expected cardinality of the result. Finally, Section 4 discusses limitations of the expected cardinality criterion, raising interesting issues on the non-unicity of solutions, and discussing other rules proposed in the literature especially when some conflict is present between the sources.

2 Preliminaries

Let X be the finite space of cardinality $|X|$ with elements $X = x_1, \ldots, x_{|X|}$.

Definition 1. *A basic belief assignment (bba) [10] is a function m from the power set of X to $[0,1]$ s.t. $m(\emptyset) = 0$ and $\sum_{A \subseteq X} m(A) = 1$.*

Let $\mathcal{M}_X$ the set of bba's on $2^{|X|}$. A set A s.t. $m(A) > 0$ is called a focal set. The number $m(A) > 0$ is the mass of A. Given a bba m, belief, plausibility and commonality functions of an event $E \subseteq X$ are, respectively

$$bel(E) = \sum_{A \subseteq E} m(A) \; ; \; pl(E) = \sum_{A \cap E \neq \emptyset} m(A) = 1 - bel(A^c) \; ; \; q(E) = \sum_{E \subseteq A} m(A)$$

A belief function measures to what extent an event is directly supported by the available information, while a plausibility function measures the maximal amount of evidence that could support a given event. A commonality function measures the quantity of mass that may be re-allocated to a particular set from its supersets. The commonality function increases when bigger focal sets receive greater mass assignments, hence the greater the commonality degrees, the less informative is the belief function. A bba is said to be non-dogmatic if $m(X) > 0$ hence $q(A) > 0, \forall A \neq \emptyset$.

A bba m can also be interpreted as a probability family [15] $\mathcal{P}_m$ such that $Bel(A)$ and $Pl(A)$ are probability bounds: $\mathcal{P}_m = \{P | \forall A \subset X, \; Bel(A) \leq P(A) \leq Pl(A)\}$. In the sequel of the paper, we mainly focus on two special kinds of bbas : namely, possibility distributions and generalized p-boxes.

A possibility distribution [16] is a mapping $\pi : X \to [0,1]$ from which two dual measures (respectively the possibility and necessity measures) can be defined : $\Pi(A) = \sup_{x \in A} \pi(x)$ and $N(A) = 1 - \Pi(A^c)$. In terms of bba, a possibility distribution is equivalent to a bba whose focal sets are nested. The plausibility (Belief) measure then reduces to a Possibility (Necessity) measure.

A p-box [9] is a pair of cumulative distributions $[\underline{F}, \overline{F}]$ defining a probability family $\mathcal{P}_{[\underline{F},\overline{F}]} = \{P | \underline{F}(x) \leq F(x) \leq \overline{F}(x) \quad \forall x \in \Re\}$. A generalized p-box [6] is a generalization of a p-box, defined on an arbitrary (especially, finite) ordered space (whereas usual p-boxes are defined on the real line). If an order $\leq_R$ is defined on X, to any bba, a generalized p-box can be associated s.t. $\overline{F}(x)_R = Pl(\{x_i | x_i \leq_R x\})$ and $\underline{F}(x)_R = Bel(\{x_i | x_i \leq_R x\})$, but it retains only a part of the information contained in the bba, generally.

Dubois and Prade [7] defined three information orderings based on different notions related to belief functions :

- pl-ordering. if $pl_1(A) \leq pl_2(A)\ \forall A \subseteq X$, we write $m_1 \sqsubseteq_{pl} m_2$;
- q-ordering. if $q_1(A) \leq q_2(A)\ \forall A \subseteq X$, we write $m_1 \sqsubseteq_q m_2$;
- s-ordering. if m_1 is a specialization of m_2, we write $m_1 \sqsubseteq_s m_2$.

Informally, a bba m_2 is a specialization of a bba m_1 if every mass $m_1(A)$ can be reallocated to subsets of A in m_2 (i.e. the mass $m_1(A)$ "flows down" to subsets $B \subseteq A$ in m_2) so as to recover m_2. If m_2 is a specialization of m_1, it means that beliefs represented by the bba m_2 are more focused than those from the bba m_1. In other words, m_2 can be judged more informative than m_1. If we interpret bbas in terms of probability families, another means to compare them in terms of imprecision is to compare such families. We can say that m_1 is more precise than m_2 iff $\mathcal{P}_{m_1} \subset \mathcal{P}_{m_2}$. This is equivalent to the pl-ordering. More generally if we have $m_1 \sqsubseteq_x m_2$ (x corresponding to one of the three orderings), we say that m_2 is x-less committed than m_1. Dubois and Prade proved that $m_1 \sqsubseteq_s m_2$ imply $m_1 \sqsubseteq_q m_2$ and $m_1 \sqsubseteq_{pl} m_2$, but that the reverse is not true (hence, s-ordering is the strongest ordering of the three).

As these relations are partial orders, comparing bbas with respect to s, pl or q-ordering can be complex and often leads to incomparability (i.e. non unicity of the solution). A simpler tool for comparing bbas is to measure the non-commitment of a bba by its expected cardinality, which reads

$$I(m) = \sum_{A \subseteq X} m(A)|A|$$

where $|A|$ is the cardinality of A. Expected cardinality is an imprecision measure, and its value is the same as the cardinality of the fuzzy set equivalent to the contour function (i.e. $I(m) = \sum_{x_i \in X} pl(x_i)$). It is coherent with specialization ordering (and hence with the two others) since if m_1 is a specialization of m_2, then $I(m_1) \leq I(m_2)$. This definition is the one we will use in the sequel.

3 A Least-Committed Merging Rule

A bba built by merging two different bbas m_1, m_2 is supposed to be obtained by the following procedure, denoting $\mathcal{F}_i$ the set of focal sets of m_i:

1. A joint bba m is built on $X \times X$, having focal sets of the form $A \times B$ where $A \in \mathcal{F}_1, B \in \mathcal{F}_2$ and preserving m_1, m_2 as marginals. It means that $m_1(A) = \sum_{B \in \mathcal{F}_2} m(A, B)$ and likewise for m_2.
2. Each joint mass $m(A, B)$ should be allocated to the subset $A \cap B$ only, where A and B are focal sets of m_1 and m_2 respectively.

We call a merging rule satisfying these two conditions *conjunctive*[1], and denote $\mathcal{M}_X^{m_1 \cap m_2}$ the set of conjunctively merged bba's. The idea behind the conjunctive

[1] A disjunctive merging rule could be defined likewise, changing $\cap$ into $\cup$.

approach is to keep as much information as possible from the fusion process. However not every bba $m_\cap$ obtained by conjunctive merging is normalized (i.e. one may get $m(\emptyset) \neq 0$). It is clear that a merged bba $m_\cap$ on X in the above sense is a specialization of both m_1 and m_2.

In fact three situations may occur

- $\mathcal{M}_X^{m_1 \cap m_2}$ contains only normalized belief functions. It means that $\forall A \in \mathcal{F}_1, B \in \mathcal{F}_2, A \cap B \neq \emptyset$. Only in that case does the result of merging by Dempster rule of combination belong to $\mathcal{M}_X^{m_1 \cap m_2}$. The two bbas are said to be *logically consistent*.
- $\mathcal{M}_X^{m_1 \cap m_2}$ contains both subnormalized and normalized bbas. It means that $\exists A, B, A \cap B = \emptyset$ and that the marginal-preservation equations have solutions which allocate zero mass $m(A, B)$ to such $A \times B$.
- $\mathcal{M}_X^{m_1 \cap m_2}$ contains only subnormalized belief functions. A result from [3] indicates that this situation is equivalent to $\mathcal{P}_{m_1} \cap \mathcal{P}_{m_2} = \emptyset$. The two bbas are said to be conflicting.

A cautious merging rule is then one that selects a least committed bba in $\mathcal{M}_X^{m_1 \cap m_2}$ for any of the three orderings given above. In order to avoid incomparabilities, we define a least-committed bba in $\mathcal{M}_X^{m_1 \cap m_2}$ as one with maximal expected cardinality $I(m)$. A conjunctive merging rule is denoted $\oplus$, and a least-committed merging rule $\bigwedge$.

Now suppose $m_1 = m_2 = m$. The least committed specialisation of m is m itself. Hence the following natural requirement:

Idempotence. The least-committed rule $\bigwedge$ should be idempotent.

The following proposition directly follows from this requirement:

Proposition 1. *Let m_1 be a specialization of m_2, then the result of the least committed rule $\bigwedge$ should be $m_1 \odot m_2 = \widehat{m}_{12} = m_1$.*

Although very important, this result concerns very peculiar cases and does not give us guidelines as to how general bbas should be combined to result in a least-committed bba (in the sense of expected cardinality). In [8], by using the concept of commensurate bbas, Dubois and Yager show that there are a lot of idempotent rules that combine two bbas, each of them giving different results. In the following, we slightly generalize the notion of bba and consider it as a relation between the power set of X and $[0, 1]$. In other words, a generalized bba may assign several weights to the same subset of X.

Definition 2. *Let m be a bba with focal sets $A_1, \ldots, A_n$ and associated weights $m^1, \ldots, m^n$. A split of m is a bba m' with focal sets $A'_1, \ldots, A'_{n'}$ and associated weights $m'^1, \ldots, m'^{n'}$ s.t. $\sum_{A'_j = A_i} m'^j = m^i$*

In other words, a split is a new bba where the original weight given to a focal set is separated in smaller weights given to the same focal set, with the sum of weights given to a specific focal set being constant. Two generalized bbas m_1,m_2

are said to be equivalent if $pl_1(E) = pl_2(E)$ and $bel_1(E) = bel_2(E)\ \forall E \subseteq X$. If m_1 and m_2 are equivalent, it means that they are splits of the same regular bba [8]. In the following, a bba should be understood as a generalized one.

Definition 3. *Let m_1, m_2 be two bbas with respective focal sets $\{A_1, \ldots, A_n\}$, $\{B_1, \ldots, B_k\}$ and associated weights $\{m_1^1, \ldots, m_1^n\}$, $\{m_2^1, \ldots, m_2^k\}$. Then, m_1 and m_2 are said to be commensurate if $k = n$ and there is a permutation σ of $\{1, \ldots, n\}$ s.t. $m_1^j = m_2^{\sigma(i)}, \forall i = 1, \ldots, n$.*

Two bbas are commensurate if their distribution of weights over focal sets can be described by the same vector of numbers. In [8], Dubois and Yager propose an algorithm, given a prescribed ranking of focal sets on each side, that makes any two bbas commensurate by successive splitting. Based on this algorithm, they provide an idempotent rule $\bigoplus$ that allows to merge any two bbas. This merging rule is conjunctive and the result depends on the ranking of focal sets used in the commensuration algorithm, summarized as follows:

- Let m_1, m_2 be two bbas and $\{A_1, \ldots, A_n\}$, $\{B_1, \ldots, B_k\}$ the two sets of ordered focal sets with weights $\{m_1^1, \ldots, m_1^n\}$, $\{m_2^1, \ldots, m_2^k\}$
- By successive splitting of each bbas (m_1, m_2), build two generalised bbas $\{R_1^1, \ldots, R_1^l\}$ and $\{R_2^1, \ldots, R_2^l\}$ with weights $\{m_{R_1}^1, \ldots, m_{R_1}^l\}$,$\{m_{R_2}^1, \ldots, m_{R_2}^l\}$ s.t. $m_{R_1}^i = m_{R_2}^i$ and $\sum_{R_1^i = A_j} = m_1^j$, $\sum_{R_2^i = B_j} = m_2^j$.
- Algorithm results in two commensurate generalised bbas m_{R_1}, m_{R_2} that are respectively equivalent to the original bbas m_1, m_2.

Once this commensuration is done, the conjunctive rule $\bigoplus$ proposed by Dubois and Yager defines a merged bba $m_{12} \in \mathcal{M}_X^{m_1 \cap m_2}$ with focal sets $\{R_{1 \bigoplus 2}^i = R_1^i \cap R_2^i, i = 1, \ldots, l\}$ and associated weights $\{m_{R_{1 \bigoplus 2}}^i = m_{R_1}^i = m_{R_2}^i, i = 1 \ldots, l\}$. The whole procedure is illustrated by the following example.

Example 1. Commensuration

m_1		m_2	
A_1	.5	B_1	.6
A_2	.3	B_2	.2
A_3	.2	B_3	.1
		B_4	.1

$\rightarrow$

l	m_{R^l}	R_1^l	R_2^l	$R_{1 \bigoplus 2}^l$
1	.5	A_1	B_1	$A_1 \cap B_1$
2	.1	A_2	B_1	$A_2 \cap B_1$
3	.2	A_2	B_2	$A_2 \cap B_2$
4	.1	A_3	B_3	$A_3 \cap B_3$
5	.1	A_3	B_4	$A_3 \cap B_4$

From this example, it is easy to see that the final result crucially depends of the chosen rankings of the focal sets of m_1 and m_2. In fact, it can be shown that any conjunctively merged bba can be produced in this way.

Definition 4. *Two commensurate generalised bbas are said to be equi-commensurate if each of their focal sets has the same weight.*

Any two bbas m_1, m_2 can be made equi-commensurate. In our example, bbas can be made equi-commensurate by splitting the first line into five similar lines

of weight 0.1 and the third line into two similar lines of weight 0.1. Every line then has weight 0.1, and applying Dubois and Yager's rule to these bbas yields a bba equivalent to the one obtained before equi-commensuration. Combining two equi-commensurate bbas $\{R_1^1, \ldots, R_1^l\}$, $\{R_2^1, \ldots, R_2^l\}$ by Dubois and Yager rule results in a bba s.t every focal element in $\{R_{1\oplus 2}^1, \ldots, R_{1\oplus 2}^l\}$ has equal weight $m_{R_{1\oplus 2}}$ (0.1 in our example). The resulting bba is still in $\mathcal{M}_X^{m_1 \cap m_2}$.

Proposition 2. *Any merged bba in $\mathcal{M}_X^{m_1 \cap m_2}$ can be reached by means of Dubois and Yager rule using appropriate commensurate bbas equivalent to m_1 and m_2 and the two appropriate rankings of focal sets.*

Proof. We assume masses (of marginal and merged bbas) are rational numbers. Let $m \in \mathcal{M}_X^{m_1 \cap m_2}$ be the merged bba we want to reach by using Dubois and Yager's rule. Let $m(A_i, B_j)$ be the mass allocated to $A_i \cap B_j$ in m. It is of the form $k_{12}(A_i, B_j) \times 10^{-n}$ where k_{12}, n are integers. By successive splitting followed by a reordering of elements R_1^j, we can always reach m. For instance, let k_R be equal to the greatest common divisor of all values $k_{12}(A_i, B_j)$. Then, $k_{12}(A_i, B_j) = q_{ij} \times k_R$, for an integer q_{ij}. Then, it suffices to re-order elements R_1^k by a re-ordering σ s.t. for q_{ij} of them, $R_1^k = A_i$ and $R_2^{\sigma(k)} = B_j$. Then, by applying Dubois and Yager's rule, we obtain the result m. From a practical standpoint, restricting ourselves to rational numbers has no importance: rational numbers being dense in reals, this means that we can always get as close as we want to any merged bba.

For cautious merging, it is natural to look for appropriate rankings of focal sets so that the merged bba obtained via commensuration has maximal cardinality. The answer is : rankings should be extensions of the partial ordering induced by inclusion (i.e. $A_i < A_j$ if $A_i \subset A_j$). This is due to the following result:

Lemma 1. *Let A, B, C, D be four sets s.t. $A \subseteq B$ and $C \subseteq D$. Then, we have the following inequality*

$$|A \cap D| + |B \cap C| \leq |A \cap C| + |B \cap D| \tag{1}$$

Proof. From the assumption, the inequality $|(B \setminus A) \cap C| \leq |(B \setminus A) \cap D|$ holds. Then consider the following equivalent inequalities:

$$\begin{aligned} |(B \setminus A) \cap C| + |A \cap C| &\leq |A \cap C| + |(B \setminus A) \cap D| \\ |B \cap C| &\leq |A \cap C| + |(B \setminus A) \cap D| \\ |A \cap D| + |B \cap C| &\leq |A \cap C| + |A \cap D| + |(B \setminus A) \cap D| \\ |A \cap D| + |B \cap C| &\leq |A \cap C| + |B \cap D| \end{aligned}$$

hence the inequality (1) is true.

When using equi-commensurate bbas, masses in the formula of expected cardinality can be factorized, and expected cardinality then becomes:
$I(m)_{R_{1\oplus 2}} = m_{R_{1\oplus 2}} \sum_{i=1}^{l} |R_{1\oplus 2}^i| = m_{R_{1\oplus 2}} \sum_{i=1}^{l} |R_1^i \cap R_2^i|$, where $m_{R_{1\oplus 2}}$ is the smallest mass enabling equi-commensuration. We are now ready to prove the following proposition.

Proposition 3. *If $m \in \mathcal{M}_X^{m_1 \cap m_2}$ is minimally committed for expected cardinality, there exists an idempotent conjunctive merging rule $\bigwedge$ constructing m by the commensuration method, s.t. focal sets are ranked on each side in agreement with the partial order of inclusion.*

Proof. Suppose $\widehat{m}_{12} \in \mathcal{M}_X^{m_1 \cap m_2}$ is minimally committed for expected cardinality. It can be obtained by commensuration. Let m_{R_1}, m_{R_2} be the two equicommensurate bbas with n elements each derived from the two original bbas m_1, m_2. Suppose that the rankings used display four focal sets $R_1^i, R_1^j, R_2^i, R_2^j$, $i < j$, such that $R_1^i \supset R_1^j$ and $R_2^i \subseteq R_2^j$. By Lemma 1, $|R_1^j \cap R_2^j| + |R_1^i \cap R_2^i| \leq |R_1^j \cap R_2^i| + |R_1^i \cap R_2^j|$. Hence, if we permute focal sets R_1^i, R_1^j before applying Dubois and Yager's merging rule, we end up with a merged bba $m_{R'_{1 \oplus 2}}$ s.t. $I(m_{R_{1 \oplus 2}}) \leq I(m_{R'_{1 \oplus 2}})$. Since any merged bba can be reached by splitting m_1,m_2 and by inducing the proper ranking of cocal sets of the resulting bbas m_{R_1}, m_{R_2}, any merged bba $\widehat{m}_{12} \in \mathcal{M}_X^{m_1 \cap m_2}$ maximizing expected cardinality can be reached by Dubois and Yager's rule, using rankings of focal sets in accordance with the inclusion ordering.

Ranking focal sets in accordance with inclusion is neither sufficient nor the only way of maximizing expected cardinality when merging two given bbas, as shown by the following examples.

Example 2. Let m_1,m_2 be two bbas of the space $X = x_1, x_2, x_3$. Let $m_1(A_1 = \{x_1, x_2\}) = 0.5$,$m_1(A_2 = \{x_1, x_2, x_3\}) = 0.5$ be the two focal sets of m_1 and $m_2(B_1 = \{x_1, x_2\}) = 0.2$,$m_2(B_2 = \{x_2\}) = 0.3$,$m_2(B_3 = \{x_1, x_2, x_3\}) = 0.5$ be the focal sets of m_2. The following table shows the result of Dubois and Yager's merging rule after commensuration:

l	m_{R^l}	R_1^l	R_2^l	$R^l_{1 \oplus 2}$
1	.2	A_1	B_1	$A_1 \cap B_1 = \{x_1, x_2\}$
2	.3	A_1	B_2	$A_1 \cap B_2 = \{x_2\}$
3	.5	A_2	B_3	$A_2 \cap B_3 = \{x_1, x_2, x_3\}$

Although focal sets B_i are not ordered by inclusion ($B_1 \supset B_2$), the result maximizes expected cardinality (the result is m_2, which is a specialization of m_1). This shows that the technique based on proposition 3 is not necessary (nevertheless, the same result is obtained by using order B_2, B_1, B_3).

Now, consider the same bba m_1 and another bba m_2 s.t. $m_2(B_1 = \{x_2\}) = 0.3$, $m_2(B_2 = \{x_2, x_3\}) = 0.3$, $m_2(B_3 = \{x_1, x_2\}) = 0.1$, $m_2(B_4 = \{x_1, x_2, x_3\}) = 0.3$. m_2 is no longer a specialization of m_1, and the order B_1, B_2, B_3, B_4 is one of the two possible extensions of the partial order induced by inclusion. The result of Dubois and Yager's rule gives us:

l	m_{R^l}	R_1^l	R_2^l	$R^l_{1 \oplus 2}$
1	.2	A_1	B_1	$A_1 \cap B_1 = \{x_2\}$
2	.3	A_1	B_2	$A_1 \cap B_2 = \{x_2\}$
3	.1	A_2	B_2	$A_2 \cap B_2 = \{x_2, x_3\}$
4	.1	A_2	B_3	$A_2 \cap B_3 = \{x_1, x_2\}$
5	.3	A_2	B_4	$A_1 \cap B_4 = \{x_1, x_2, x_3\}$

and the expected cardinality of the merged bba is 1.8. If, instead of the order B_1, B_2, B_3, B_4, we choose the order B_1, B_3, B_2, B_4 (i.e. the other extension of the partial order induced by inclusion), applying Dubois and Yager's rule gives us a merged bba of expected cardinality 2.0, which is higher than the previous one. Hence, we see that proposition 3 is not sufficient in general to reach maximal cardinality. Thus, proposition 3 gives us guidelines for combining belief functions so as to maximise cardinality, but further conditions should be stated to select the proper total orderings of focal sets.

4 Beyond Least-Commitment Based on Expected Cardinality

Least committed merging by expected cardinality maximisation is coherent with specialization since if an s-least committed bba exists, then it has maximal expected cardinality. But other notions of minimal commitment exist, that do not relate to expected cardinality. This section discusses arguments pro and con the use of this notion, first for logically consistent bbas and then for more general ones.

4.1 Retrieving the Minimum Rule of Possibility Theory

For the special case of possibility distributions, the order between focal sets induced by inclusion is complete. It means that, in this case, applying proposition 3 results in an unique consonant merged bba with contour function $\min(\pi_1, \pi_2)$, which corresponds to the usual minimum operator [8]. As the minimum is the most cautious conjunctive merging operator in possibility theory, it shows that our proposition is coherent with and thus justifies the possibilistic approach, as suggested by Smets[12]. One may also conjecture that merged bbas that maximize expected cardinality are also least-committed in the sense of the relative specificity of their contour functions (m_1 is less committed than m_2 in this sense if $pl_1(x) \geq pl_2(x)\ \forall x \in X$). Nevertheless, the minimum of two possibility distributions is not the only cardinality maximizer, as the next example shows:

Example 3. Consider the two following possibility distributions π_1,π_2, expressed as belief structures m_1, m_2

$\pi_1 = m_1$		$\pi_2 = m_2$	
Focal sets	Mass	Focal sets	Mass
$\{x_1, x_2, x_3\}$	0.5	$\{x_3, x_4, x_5\}$	0.5
$\{x_0, x_1, x_2, x_3, x_4\}$	0.5	$\{x_2, x_3, x_4, x_5, x_6\}$	0.5

The following merged bbas $C_1, C_2 \in \mathcal{M}_X^{m_1 \cap m_2}$ have the same contour function, hence (maximal) expected cardinality equal to 2.

$C_1 = \pi_{min}$		C_2	
Focal sets	Mass	Focal sets	Mass
$C_{11} = \{x_3\}$	0.5	$C_{21} = \{x_3, x_4\}$	0.5
$C_{12} = \{x_2, x_3, x_4\}$	0.5	$C_{22} = \{x_2, x_3\}$	0.5

This interesting example is discussed below.

4.2 Refining Expected Cardinality by the pl- or q-Ordering

As maximizing expected cardinality is coherent with s-least commitment and can lead to non-uniqueness of the solution, discriminating different solutions can be done by using pl- or q-ordering. Choosing one or the other matters, since even for the simple example 3, we have $C_1 \sqsubset_{pl} C_2$ and $C_2 \sqsubset_q C_1$. Since $C_1 \sqsubset_{pl} C_2$ is equivalent to $\mathcal{P}_{C_1} \subset \mathcal{P}_{C_2}$ (e.g. the probability distribution $p(x_2) = 0.5, p(x_4) = 0.5$ is inside $\mathcal{P}_{C_2}$, and not in $\mathcal{P}_{C_1}$), choosing the *pl* ordering is coherent with a probabilistic interpretation of belief functions and shows the limitation of proposition 3. Note that in the example, the bba C_2 is the generalized p-box (with the order $x_1 <_R x_2 <_R \ldots <_R x_n$ on elements of X) corresponding to the possibility distribution C_1. It is not surprising that $\mathcal{P}_{C_1} \subset \mathcal{P}_{C_2}$, since the probability family induced by a possibility distribution is included in the family induced by its corresponding p-box [1].

Besides, choosing the $q-$ordering to discriminate solutions (which yields C_1 in example 3) seems more in accordance with proposition 3 (and thus with the particular case of possibility distributions). Moreover, as the commonality function increases when larger focal sets receive greater mass assignments, it could be argued that the $q-$ordering is more in accordance with the TBM approach. Smets [12] suggests without proof that in the case of merging possibility distributions, the minimum rule is least $q-$committed, like in the example.

4.3 Minimizing Conflict

When two bbas are not logically consistent (i.e. there are focal elements A_i,B_j for which $A_i \cap B_j = \emptyset$), a conjunctively merged bba that maximizes expected cardinality may not, in general, minimize conflict (i.e. $m \in \mathcal{M}_X^{m_1 \cap m_2}$ s.t. $m(\emptyset)$ is minimal). This is illustrated by the following example:

Example 4. Consider the two following possibility distributions π_1,π_2, expressed as belief structures m_1, m_2

$\pi_1 = m_1$		$\pi_2 = m_2$	
Focal sets	Mass	Focal sets	Mass
$\{x_1, x_2\}$	0.5	$\{x_4\}$	0.5
$\{x_0, x_1, x_2, x_3, x_4\}$	0.5	$\{x_2, x_3, x_4, x_5, x_6\}$	0.5

And the following table shows the result of applying the minimum (thus maximising expected cardinality) and the unnormalized Dempster rule of combination

Min(π_1, π_2)		unnormalized Dempster's rule			
Focal sets	Mass	Focal sets	Mass	Focal sets	Mass
$\{x_2, x_3, x_4\}$	0.5	$\{x_2\}$	0.25	$\{x_2, x_3, x_4\}$	0.25
$\emptyset$	0.5	$\{x_4\}$	0.25	$\emptyset$	0.25

With Dempster rule, conflict value is 0.25 and expected cardinality is 1.25, while with the minimum, the conflict value is 0.5 and expected cardinality is 1.5.

Provided one considers that minimizing the conflict is as desirable as finding a least-committed way of merging the information, this can problematic. A possible alternative is then to find $m \in \mathcal{M}_X^{m_1 \cap m_2}$ that is least-committed among those for which $m(\emptyset)$ is minimal. This problem was studied by Cattaneo in [2]. Cattaneo proposes to find the merged bba $m \in \mathcal{M}_X^{m_1 \cap m_2}$ that maximizes the following function:

$$F(m) = m(\emptyset)f(0) + (1 - m(\emptyset)) \sum_{A \neq \emptyset} m(A) log_2(A) \tag{2}$$

with $f(0)$ a real number s.t. $f(0) < |X|$. In the above equation, $m(\emptyset)f(0)$ can be seen as a penalty given to the evaluation of the merged belief when conflict appears, while the second part of the right-hand side of equation (2) is equivalent to expected cardinality where $|A|$ is replaced by $log_2(|A|)$ (more generally, we can replace $|A|$ by any non-decreasing function $f(|A|)$ from $\mathbb{N}$ to $\mathbb{R}$). A similar strategy (penalizing the appearance of conflict) could thus be adopted with expected cardinality (or with any function $f(|A|)$), nevertheless, it would not be without inconvenient:

- adding penalty to conflict is computationally less efficient than using expected cardinality alone, since proposition 3 does not hold.
- Cattaneo mentions that associativity and conflict minimization are incompatible, while our rule is at least associative in the case of possibility distributions (other cases still have to be explored).

Now, the claim that a cautious conjunctive rule should give a merged bba where the conflict is minimized is questionable. This is shown by our small example 4, where minimizing the conflict, by assigning zero mass to empty intersections while respecting the marginals, produces the bba $m(\{x_2\}) = 0.5, m(\{x_4\}) = 0.5$, which is the only probability distribution distribution in $\mathcal{P}_{m_1} \cap \mathcal{P}_{m_2}$. Indeed, this bba is the most precise possible result, and its informational content is clearly more adventurous than the bba corresponding to $\min(\pi_1, \pi_2)$.

4.4 Least Commitment Based on the Weight Function

Any non dogmatic belief function with bba m can be uniquely represented as the conjunctive combination of the form $m = \bigodot_{A \neq X} A^{w(A)}$ [11], where $w(A)$ is a positive weight, and $A^{w(A)}$ represents the (generalized) simple support function with bba μ such that $\mu(A) = 1 - w(A)$ and $\mu(X) = w(A)$, and $\bigodot$ denotes the unnormalized Dempster rule of combination. Note that if $w(A) \in [0, 1]$, μ is a simple support bba. Otherwise, it is not a bba. Denoeux [5] introduces another definition of least commitment calling a bba m_1 less w-committed than m_2 whenever $w_1(A) \leq w_2(A), \forall A \neq X$. Denoeux proposes to apply the following cautious rule to weight functions:

$$w_{12}(A) = \min(w_1(A), w_2(A)), \forall A \neq X.$$

and he shows that it produces the weight function of the least w-committed merged bba among those that are more w-committed than both marginals m_1 and m_2. If a bba is less w-committed than another one, then it is a specialisation thereof. Our conjunctive merging only requires the result to be more s-committed than m_1 and m_2. which is a weaker condition than to be more w-committed. Now, apply both rules to the following example (2 of [5]).

Example 5. : Consider $X = \{a, b, c\}$, m_1 defined by $m_1(\{a, b\}) = 0.3$, $m_1(\{b, c\}) = 0.5$, $m_1(X) = 0.2$; m_2 defined by $m_2(\{b\}) = 0.3$, $m_2(\{b, c\}) = 0.4$, $m_2(X) = 0.3$. Results of both rules are given in the following table

Denoeux's rule (m^D)				Max. Exp. Card. rule (m^C)			
Focal Sets	Mass	Focal Sets	Mass	Focal Sets	Mass	Focal Sets	Mass
$\{b\}$	0.6	$\{b, c\}$	0.2	$\{b\}$	0.3	$\{X\}$	0.2
$\{a, b\}$	0.12	$\{X\}$	0.08	$\{b, c\}$	0.5		

In this example, our conjunctive cautious rule yields a merged bba m^C that is s-less committed (and hence has a greater expected cardinality) than m^D, the one obtained with Denoeux's rule. Nevertheless, the merged bba obtained by maximizing expected cardinality is not comparable in the sense of the w-ordering with any of the three other bbas (m_1, m_2, m^D), nor does it fulfil Denoeux's condition of being more w-committed than m_1 and m_2. The cautious w-merging of possibility distributions does not reduce to the minimum rule either. Thus, the two approaches are at odds. As it seems, using the w-ordering allows to easily find a unique least-committed element, at the expense of restricting the search to a subset of $\mathcal{M}_X^{m_1 \cap m_2}$ due to the use of the w-ordering (which can be questioned in the scope of a cautious approach). See [5] for a more detailed discussion on these issues.

In his paper, Denoeux generalizes both $\odot$ and his cautious rule with triangular norms. However, the set of non-dogmatic belief functions equipped with $\odot$ forms a group, as is the product of positive w-numbers. So the relevant setting for generalizing the product of weight functions seems to be the one of uninorms. But the minimum is not a uninorm on the positive real line. It is the greatest t-norm on $[0, 1]$, in particular, greater than product, and this property is in agreement with minimal commitment of contour functions. But the minimum rule no longer dominates the product on the positive real line, so that the bridge between Denoeux's idempotent rule and the idea of minimal commitment is not obvious beyond the w-ordering.

5 Conclusions

When our knowledge about the dependencies existing between multiple sources is poor, Dempster rule of combination cannot be applied. The merging of bbas should follow the principle of least-commitment, or said differently, we should adopt a cautious attitude. Nevertheless, the various definitions of least-commitment often lead to indecision (i.e. to non-unicity of the solution). In this paper, we have studied the maximisation of the expected cardinality of the merged bba and proposed

an idempotent merging rule, based on the commensuration of bbas respecting the partial ordering induced by inclusion between focal sets. It encompasses the minimum rule on possibility distributions, thus justifying it in terms of least commitment. However more investigations are needed to make our proposition practically convenient and to articulate the expected cardinality criterion with other notions of least commitment, based on generalized forms of bba cardinality, on the comparison of contour functions, and other information orderings in the theory of belief functions.

References

1. Baudrit, C., Dubois, D.: Practical representations of incomplete probabilistic knowledge. Computational Statistics and Data Analysis 51(1), 86–108 (2006)
2. Cattaneo, M.: Combining belief functions issued from dependent sources. In: Proc. Third International Symposium on Imprecise Probabilities and Their Application (ISIPTA'03), Lugano, Switzerland, pp. 133–147 (2003)
3. Chateauneuf, A.: Combination of compatible belief functions and relation of specificity. In: Advances in the Dempster-Shafer theory of evidence, pp. 97–114. John Wiley & Sons, New York (1994)
4. Dempster, A.: Upper and lower probabilities induced by a multivalued mapping. Annals of Mathematical Statistics 38, 325–339 (1967)
5. Denoeux, T.: Conjunctive and disjunctive combination of belief functions induced by non distinct bodies of evidence. Artificial Intelligence (to appear)
6. Destercke, S., Dubois, D.: A unified view of some representations of imprecise probabilities. In: Lawry, J., Miranda, E., Bugarin, A., Li, S. (eds.) Int. Conf. on Soft Methods in Probability and Statistics (SMPS), Advances in Soft Computing, Bristol, pp. 249–257. Springer, Heidelberg (2006)
7. Dubois, D., Prade, H.: A set-theoretic view on belief functions: logical operations and approximations by fuzzy sets. I. J. of General Systems 12, 193–226 (1986)
8. Dubois, D., Yager, R.: Fuzzy set connectives as combination of belief structures. Information Sciences 66, 245–275 (1992)
9. Ferson, S., Ginzburg, L., Kreinovich, V., Myers, D., Sentz, K.: Constructing probability boxes and dempster-shafer structures. Technical report, Sandia National Laboratories (2003)
10. Shafer, G.: A mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
11. Smets, P.: The canonical decomposition of a weighted belief. In: Proc. Int. Joint, Montreal, pp. 1896–1901 (1995)
12. Smets, P.: Quantified possibility theory seen as an hyper cautious transferable belief model. In: Proceedings LFA2000, La Rochelle. Cepadues-Editions Toulouse, pp. 343–353 (2000)
13. Smets, P.: Analyzing the combination of conflicting belief functions. Information Fusion 8, 387–412 (2007)
14. Smets, P., Kennes, R.: The transferable belief model. Artificial Intelligence 66, 191–234 (1994)
15. Walley, P.: Statistical reasoning with imprecise Probabilities. Chapman and Hall, Sydney (1991)
16. Zadeh, L.: Fuzzy sets as a basis for a theory of possibility. Fuzzy sets and systems 1, 3–28 (1978)

Consonant Belief Function Induced by a Confidence Set of Pignistic Probabilities

Astride Aregui[1,2] and Thierry Denœux[1]

[1] HEUDIASYC, UTC, CNRS
Centre de Recherche de Royallieu
BP 20529, F-60205 Compiègne, France
[2] CIRSEE, Suez Environnement
38 rue du Pdt Wilson, F-78230 Le Pecq, France

Abstract. A new method is proposed for building a predictive belief function from statistical data in the Transferable Belief Model framework. The starting point of this method is the assumption that, if the probability distribution $\mathbb{P}_X$ of a random variable X is known, then the belief function quantifying our belief regarding a future realization of X should have its pignistic probability distribution equal to $\mathbb{P}_X$. When $\mathbb{P}_X$ is unknown but a random sample of X is available, it is possible to build a set $\mathscr{P}$ of probability distributions containing $\mathbb{P}_X$ with some confidence level. Following the Least Commitment Principle, we then look for a belief function less committed than all belief functions with pignistic probability distribution in $\mathscr{P}$. Our method selects the most committed consonant belief function verifying this property. This general principle is applied to the case of the normal distribution.

Keywords: Dempster-Shafer theory, Evidence theory, Transferable Belief Model, possibility distribution, statistical data.

1 Introduction

The Transferable Belief Model (TBM) is gaining increasing interest as a formal framework for information fusion, decision making under uncertainty and imprecise data analysis [14,21,18]. However, it is no always clear how to quantify various uncertainties using belief functions as required in this framework, especially when statistical data are involved. A contribution to this problem will be presented here.

More precisely, the problem considered in this paper can be described as follows. Let X be a random variable with unknown probability distribution $\mathbb{P}_X$. We would like to quantify the beliefs held by an agent about a future realization of X from past independent observations $X_1, \ldots, X_n$ drawn from the same distribution. In [5], it was argued that a belief function $bel(\cdot; X_1, \ldots, X_n)$ solution to this problem should verify two properties: it should be less committed than $\mathbb{P}_X$ with a given probability (i.e., for a given proportion of realizations of the random sample), and it should converge towards $\mathbb{P}_X$ in probability as the size of the sample tends to infinity. Several methods for constructing such belief functions (refered to as *predictive belief functions*) were proposed in [5] in the special case where X is discrete, based on multinomial confidence intervals. This approach was

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 344–355, 2007.

recently extended to the continuous case using confidence bands on the unknown cumulative probability distribution instead of multinomial confidence intervals [1], and a similar approach in the context of Possibility Theory was presented in [12].

In the above approach, the second requirement demanding that, in the long run, the predictive belief function converge towards the probability distribution of X is based on Hacking's frequency principle [11,17], which equates the degree of belief of an event to its probability (long run frequency), when the latter is known. This principle, however, can be questioned. For instance, consider the result X of a coin-tossing experiment, with $X \in \{H, T\}$, where H and T stand for "Head" and "Tail", respectively. If the coin is known to be perfectly balanced, then $\mathbb{P}_X(\{H\}) = \mathbb{P}_X(\{T\}) = 0.5$. If asked about our opinion regarding the result of the next toss, should we necessarily assign a degree of belief 0.5 to the event that this toss will bring a "Head"? This requirement seems hard to justify. However, if we are forced to bet on the result of this random experiment, then it seems reasonable to assign equal odds to the two elementary events. In the TBM, degrees of chance are not equated with degrees of belief: decision making is assumed to be handled at the *pignistic level*, which is distinguished from the *credal level* at which beliefs are entertained [21,20]. The pignistic transformation converts each belief function *bel* into a *pignistic* probability distribution *BetP* that is used for decision making. As a consequence, we may replace Hacking's principle by the weaker requirement that the pignistic probability of an event be equal to its long run frequency, when the latter is known. Coming back to the coin example, this requirement leads to the constraint $BetP(\{H\}) = BetP(\{T\}) = 0.5$, which defines a set of admissible belief functions. Among this set, the Least Commitment Principle [16] dictates to choose the least committed one (i.e., the least informative), which is here the vacuous belief function.

In the above example, the probability distribution of X was assumed to be known. In the more realistic situation considered here, we only have partial information about this distribution, in the form of a random sample $X_1, \ldots, X_n$. In that case, it is possible to construct a set $\mathscr{P}$ of probability distributions defined, e.g., by a parametric confidence region. A natural extension of the above line of reasoning is then to require that *bel* be less committed than any belief function with pignistic probability distribution in $\mathscr{P}$. This leads to the definition of a set of admissible belief functions, among which the most committed one can be chosen. This is the principle of the approach presented in this paper.

The rest of this paper is organized as follows. The background on the TBM will first be recalled in Section 2. The proposed approach will be formalized in Section 3. It will then be applied to the case of the normal distribution in Section 4. Section 5 will finally conclude the paper.

2 Background on the TBM

This section provides a short introduction to the main notions pertaining to the theory of belief functions that will be used throughout the paper, and in particular, its TBM interpretation. We first consider the case of belief functions defined on a finite domain [14], and then address the case of a continuous domain [19].

2.1 Belief Functions on a Finite Domain

Let $\mathscr{X} = \{\xi_1, \ldots, \xi_K\}$ be a finite set, and let X be a variable taking values in $\mathscr{X}$. Given some evidential corpus, the knowledge held by a given agent at a given time over the actual value of variable X can be modeled by a so-called *basic belief assignment* (bba) m defined as a mapping from $2^{\mathscr{X}}$ into $[0, 1]$ such that:

$$\sum_{A \subseteq \mathscr{X}} m(A) = 1. \tag{1}$$

Each mass $m(A)$ is interpreted as the part of the agent's belief allocated to the hypothesis that X takes some value in A [14,21]. The subsets $A \in \mathscr{X}$ such that $m(A) > 0$ are called the focal sets of A. When the focal sets are nested, m is said to be consonant.

Equivalent representations of m include the belief, plausibility and commonality functions defined, respectively, as:

$$bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B), \tag{2}$$

$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \tag{3}$$

and

$$q(A) = \sum_{B \cap A \neq \emptyset} m(B), \tag{4}$$

for all $A \subseteq \mathscr{X}$. When m is consonant, then the plausibility function is a possibility measure: it verifies $pl(A \cup B) = \max(pl(A), pl(B))$ for all $A, B \subseteq \mathscr{X}$. The corresponding possibility distribution is defined by $\text{poss}(x) = pl(\{x\}) = q(\{x\})$ for all $x \in \mathscr{X}$, and the commonality function verifies $q(A \cup B) = \min(q(A), q(B))$ for all $A, B \subseteq \mathscr{X}$. Conversely, any possibility measure Π with possibility distribution $\text{poss}(x) = \Pi(\{x\})$ for all $x \in \mathscr{X}$ is a plausibility function corresponding to a consonant bba m defined as follows [7]. Let $\pi_k = \text{poss}(\xi_k)$, and let us assume that the elements of $\mathscr{X}$ have been arranged in such a way that $\pi_1 \geq \pi_2 \geq \ldots \geq \pi_K$. Then, we have:

$$m(A) = \begin{cases} 1 - \pi_1 & \text{if } A = \emptyset, \\ \pi_k - \pi_{k+1} & \text{if } A = \{\xi_1, \ldots, \xi_k\} \text{ for some } k \in \{1, \ldots, K-1\}, \\ \pi_K & \text{if } A = \mathscr{X}, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

In the TBM, the *Least commitment Principle* (LCP) plays a role similar to the principle of maximum entropy in Bayesian Probability Theory. As explained in [16], the LCP states that, given two belief functions compatible with a set of constraints, the most appropriate is the least informative. To make this principle operational, it is necessary to define ways of comparing belief functions according to their information content. Several such partial orderings, generalizing set inclusion, have been proposed [22,8]. Among them, the q- and pl-ordering relations are defined as follows:

- m_1 is said to be q-more committed than m_2 (noted $m_1 \sqsubseteq_q m_2$) if $q_1(A) \leq q_2(A)$, for all $A \subseteq \mathscr{X}$;

– m_1 is said to be *pl*-more committed than m_2 (noted $m_1 \sqsubseteq_{pl} m_2$) if $pl_1(A) \leq pl_2(A)$, for all $A \subseteq \mathscr{X}$;

The interpretation of these and other ordering relations is discussed in [8] from a set-theoretical perspective, and in [9] from the point of view of the TBM. In general, *q*- and *pl*-orderings are distinct notions, and none of them implies the other. However, these two orderings are equivalent in the special case of consonant belief functions: if m_1 and m_2 are consonant, then

$$m_1 \sqsubseteq_q m_2 \Leftrightarrow m_1 \sqsubseteq_{pl} m_2 \Leftrightarrow \text{poss}_1 \leq \text{poss}_2.$$

The TBM is a two-level mental model in which belief representation and updating take place at a first level termed *credal level*, whereas decision making takes place at a second level called *pignistic level* [21]. To make decisions, any bba m such that $m(\emptyset) < 1$ is mapped into a pignistic probability function $Betp = Bet(m)$ given by

$$Betp(x) = \sum_{A \subseteq \mathscr{X}, A \neq \emptyset} \frac{m(A)}{1 - m(\emptyset)} \frac{1_A(x)}{|A|}, \quad \forall x \in \mathscr{X}, \tag{6}$$

where 1_A denotes the indicator function of A defined by $1_A(x) = 1$ if $x \in A$, 0 otherwise.

Conversely, let us assume that we know the pignistic probability function p_0 of an agent and we would like to find the *q-least committed* (*q*-LC) belief function associated to p_0. As shown in [9,10], the solution is a consonant belief function, called the *q*-LC *isopignistic* belief function. It is defined by the following possibility distribution:

$$\text{poss}(x) = \sum_{x' \in \mathscr{X}} \min(p_0(x), p_0(x')). \tag{7}$$

If m is the bba associated to poss, we note $m = Bet_{LC}^{-1}(p_0)$.

2.2 Continuous Belief Functions on $\mathbb{R}$

Belief functions on $\mathbb{R}$ may be defined by replacing the concept of bba by that of basic belief density (bbd) [4,15,19]. A normal bbd m is a function taking values from the set of closed real intervals into $[0, +\infty)$, such that

$$\iint_{x \leq y} m([x, y])\, dx\, dy = 1. \tag{8}$$

The belief, plausibility and commonality functions can be defined in the same way as in the finite case, replacing finite sums by integrals. In particular,

$$bel([x, y]) = \int_x^y \int_u^y m([u, v]) dv du, \tag{9}$$

$$pl([x, y]) = \int_{-\infty}^y \int_{\max(x,u)}^{+\infty} m([u, v]) dv du, \tag{10}$$

$$q([x,y]) = \int_{-\infty}^{x} \int_{y}^{+\infty} m([u,v])dvdu, \tag{11}$$

for all $x \leq y$. The domains of these integrals may be represented as in Figure 1, where each point in the triangle corresponds to an interval with upper and lower bounds indicated on the horizontal and vertical axes, respectively.

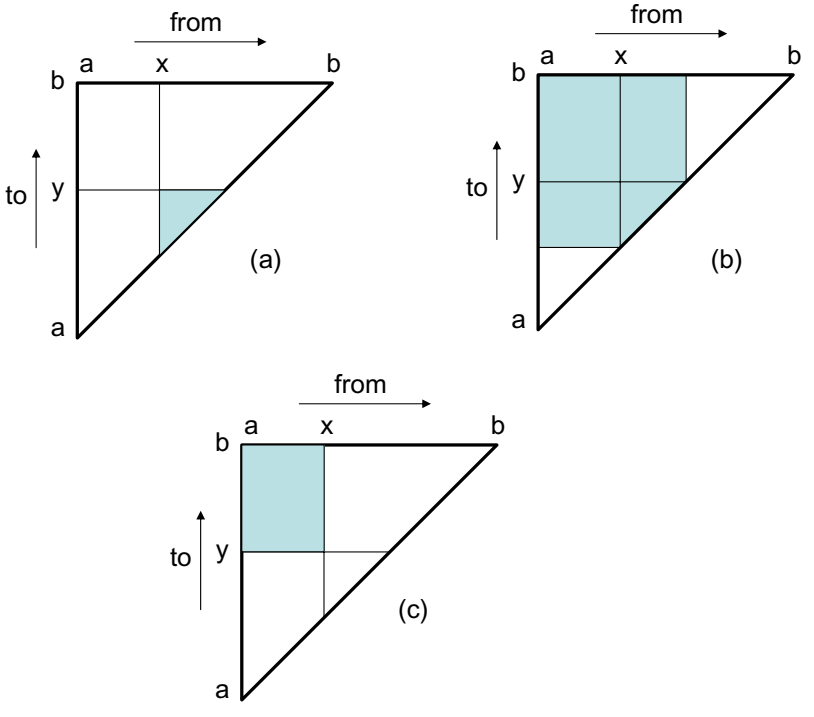


Fig. 1. The belief, plausibility and commonality functions are defined as integrals of the bbd with support $[a, b]$ on the shaded area of triangles (a), (b) and (c), respectively

A pignistic probability distribution $Betf = Bet(m)$ can be defined as in the discrete case. It is a continuous distribution with the following probability density [19]:

$$Betf(x) = \lim_{\epsilon \to 0} \int_{-\infty}^{x} \int_{x+\epsilon}^{+\infty} \frac{m([u,v])}{v-u} dvdu. \tag{12}$$

The expression of the q-LC isopignistic bbd $m = Bet_{LC}^{-1}(f_0)$ associated with a unimodal probability density f_0 with mode ν was also derived in [19]. The focal sets of m are the level sets of the density function f_0. They are intervals $I_b = [a, b]$ such that $f_0(a) = f_0(b)$. Given the upper bound b of any such interval, the lower bound is uniquely defined by $a = \gamma(b)$ for all $b \geq \nu$. The bbd is defined by

$$m([a,b]) = \theta(b)\delta(a - \gamma(b)),$$

with

$$\theta(b) = (\gamma(b) - b) f_0'(b),$$

where f_0' is the derivative of f_0 and δ is the Dirac delta function. Note that m is consonant. Consequently, the associated plausibility function is a possibility measure. The corresponding possibility distribution poss is given by:

$$\text{poss}(x) = pl(\{x\}) = \begin{cases} \int_x^{+\infty} (\gamma(t) - t) f_0'(t) dt & \text{if } x \geq \nu \\ \int_{\gamma^{-1}(x)}^{+\infty} (\gamma(t) - t) f_0'(t) dt \text{ otherwise.} \end{cases}$$

If f_0 is symmetrical, then $\gamma(x) = 2\nu - x$, and the above equation simplifies to

$$\text{poss}(x) = \begin{cases} 2(x - \nu) f_0(x) + 2 \int_x^{+\infty} f_0(t) dt \text{ if } x \geq \nu \\ 2(\nu - x) f_0(x) + 2 \int_{-\infty}^{x} f_0(t) dt \text{ otherwise.} \end{cases} \tag{13}$$

3 Consonant Belief Function Induced by a Set of Pignistic Probabilities

Let us now assume that the pignistic probability distribution p_0 of an agent is only known to belong to a set $\mathscr{P}$ of probability distributions and, as before, we seek to approximate the agent's bba m_0. The problem is again underdetermined, as we can only say that m_0 belongs to the set $\mathscr{M}(\mathscr{P}) = Bet^{-1}(\mathscr{P})$ defined by

$$\begin{aligned} \mathscr{M}(\mathscr{P}) &= \{m \mid Bet(m) \in \mathscr{P}\} \\ &= \bigcup_{p \in \mathscr{P}} \mathscr{M}(p), \end{aligned}$$

where $\mathscr{M}(p) = Bet^{-1}(p)$ denotes the set of bbas whose pignistic probability distribution is equal to p (see Figure 2).

According to the LCP, m_0 should be approximated by a bba m^* less committed than m_0, with respect to some ordering $\sqsubseteq$. In general, the set $\mathscr{M}(\mathscr{P})$ does not contain a LC element. However, we may define the *admissible* set $\mathscr{M}^*(\mathscr{P})$ as the set of bbas *dominating* (i.e., less committed than) all bbas in $\mathscr{M}(\mathscr{P})$:

$$\mathscr{M}^*(\mathscr{P}) = \{m' \mid m \sqsubseteq m', \forall m \in \mathscr{M}(\mathscr{P})\}.$$

It is then natural to choose m^* as the *most committed* element in $\mathscr{M}^*(\mathscr{P})$, if this element exists. The solution of this problem is not obvious in the general case. However, a simple solution can be found if we restrict the search to the subset $\mathscr{C}^*(\mathscr{P}) \subset \mathscr{M}^*(\mathscr{P})$ of *consonant* bbas less committed than all bbas in $\mathscr{M}(\mathscr{P})$, and we consider the q-ordering.

For all $p \in \mathscr{P}$, let $m_p = Bet_{LC}^{-1}(p)$ be the q-LC isopignistic bba induced by p. It is consonant. Let poss_p denote the corresponding possibility distribution. Bba m_p is the q-least committed bba in the set $\mathscr{M}(p)$ of bbas whose pignistic probability distribution is p. Consequently, a consonant bba m belongs to $\mathscr{C}^*(\mathscr{P})$ if and only if it is q-less committed than m_p, for all $p \in \mathscr{P}$, ie, if and only if

$$\text{poss}_p \leq \text{poss}, \quad \forall p \in \mathscr{P},$$

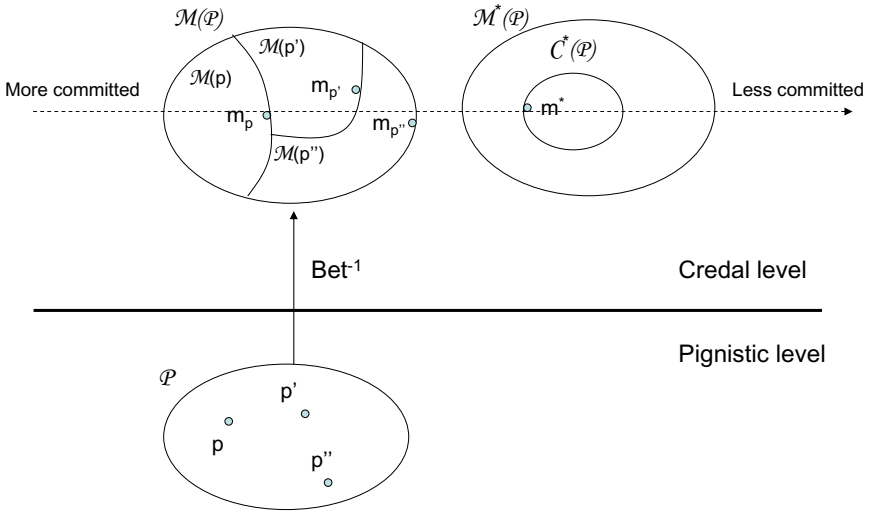


Fig. 2. Definition of the q-most committed dominating (q-MCD) bba m^* associated to a set $\mathscr{P}$ of probability distribution. The set $\mathscr{M}(\mathscr{P})$ contains all bbas with pignistic probability function in $\mathscr{P}$. The set $\mathscr{M}^*(\mathscr{P})$ contains all bbas dominating (i.e., less committed than) all bbas in $\mathscr{M}(\mathscr{P})$. The q-MCD bba m^* is the q-most committed consonant bba in $\mathscr{M}^*(\mathscr{P})$.

where poss is the possibility distribution associated to m. It follows that the q-most committed element in $\mathscr{C}^*(\mathscr{P})$ is defined by the following possibility distribution

$$\text{poss}^*(x) = \sup_{p \in \mathscr{P}} \text{poss}_p(x), \quad \forall x \in \mathscr{X}. \tag{14}$$

Possibility distribution poss* will be referred to as the *q-most committed dominating* (q-MCD) possibility distribution associated to $\mathscr{P}$. The corresponding bba will be noted m^*.

Example 1. Let us consider a frame $\mathscr{X} = \{\xi_1, \xi_2, \xi_3\}$ with three elements, and a set $\mathscr{P} = \{p, p', p''\}$ of three probability distributions shown in the first three columns of Table 1. The possibility distributions poss, poss$'$, poss$''$ associated with the corresponding q-LC isopignistic bbas are displayed in Table 1. Note that there is no q-LC element among these three bbas. Possibility distribution poss* is shown in the last column of Table 1. Using (5), we obtain the corresponding bba as

$$m^*(\{\xi_1\}) = 0.35, \quad m^*(\{\xi_1, \xi_2\}) = 0.05, \quad m^*(\mathscr{X}) = 0.6.$$

Remark 1. By definition, the q-MCD bba m^* is the q-most committed element among all *consonant* bbas that are q-less committed than all bbas in $\mathscr{M}(\mathscr{P})$. The restriction to consonant bbas is justified by the existence and unicity of a solution in $\mathscr{C}^*(\mathscr{P})$, whereas the existence of a q-most committed element in $\mathscr{M}^*(\mathscr{P})$ is not guaranteed in general. Additionally, finding the solution in $\mathscr{C}^*(\mathscr{P})$ is computationally tractable in

Table 1. Pignistic probabilities and corresponding q-LC isopignistic possibility distributions of Example 1

x	$p(x)$	$p'(x)$	$p''(x)$	poss(x)	poss$'(x)$	poss$''(x)$	poss$^*(x)$
ξ_1	0.7	0.6	0.65	1	1	1	1
ξ_2	0.2	0.25	0.1	0.5	0.65	0.3	0.65
ξ_3	0.1	0.15	0.25	0.3	0.45	0.6	0.6

several cases of practical interest, as will be shown below, and the result usually has a very simple expression. It may happen, however, that a q-most committed element in $\mathscr{M}^*(\mathscr{P})$ exists, and that it is strictly more committed than m^*. This is the case, in particular, when function q_{max} defined by

$$q_{max}(A) = \max_{p \in \mathscr{P}} q_p(A), \quad \forall A \subseteq \mathscr{X}$$

is a commonality function, q_p being the commonality function associated to m_p, . In that case, the corresponding bba m_{max} is obviously the q-most committed element in $\mathscr{M}^*(\mathscr{P})$. This is the case in Example 1: it may be shown that $q_{max} = \max(q, q', q'')$ is a commonality function, and the corresponding bba m_{max} is strictly q-more committed than m^*.

Remark 2. The approach presented here is different from that introduced in [5] and [2], in which we searched for the *pl*-most committed bba m°, in the set $\mathscr{M}^\circ(\mathscr{P})$ of bbas that are less committed than *all probability measures* in $\mathscr{P}$. In this alternative approach, the solution is obtained as the lower envelope P_* of $\mathscr{P}$, when it is a belief function. This is the case, in particular, when $\mathscr{P}$ is a p-box [2], or when it is constructed from a multinomial confidence region with $K \leq 3$ [5]. Different heuristics were introduced in [5] for constructing a belief function less committed than P_* when P_* is not a belief function. The approach adopted here usually yields a simpler result as it produces consonant belief functions. Additionally, it may be argued to be more in line with the two-level structure of the TBM, as it does not directly compare probabilities at the pignistic level with belief functions at the credal level.

4 Application to the Normal Distribution

Let us now assume that X has a normal distribution with mean μ and variance σ^2. If these two parameters are known, then the possibility distribution poss associated with the q-LC isopignistic bbd is given by (13):

$$\text{poss}(x; \mu, \sigma) = \begin{cases} \frac{2(x-\mu)}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) + 2\left(1 - \Phi\left(\frac{x-\mu}{\sigma}\right)\right) & \text{if } x \geq \mu \\ \frac{2(\mu-x)}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) + 2\Phi\left(\frac{x-\mu}{\sigma}\right) & \text{otherwise,} \end{cases} \tag{15}$$

where Φ is the standard normal cumulative distribution function.

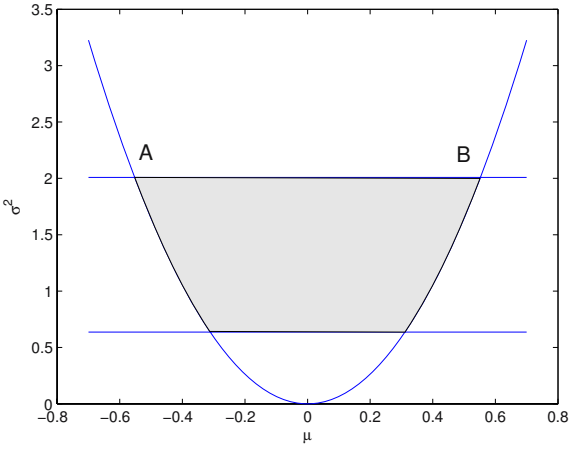


Fig. 3. Shape of Mood's exact region: the Mood Exact Region for $\alpha = 0.1$, $\alpha_1 = \alpha_2$ and $n = 25$. Without loss of generality, $\overline{x} = 0$ and $s^2 = 1$. The points with coordinates $(\widehat{\mu^-}, (\widehat{\sigma^+})^2)$ and $(\widehat{\mu^+}, (\widehat{\sigma^+})^2)$ are denoted A and B, respectively.

When μ and σ^2 are unknown but an iid sample $X_1, \ldots, X_n$ is available, then it is possible to define a joint confidence region for μ and σ^2 [3]. In particular, the Mood exact confidence region at level $1 - \alpha = (1 - \alpha_1)(1 - \alpha_2)$ is defined by

$$\mathscr{R}(X_1, \ldots, X_n) = \left\{ (\mu; \sigma^2) : \overline{X} - u_{1-\alpha_1/2} \frac{\sigma}{\sqrt{n}} \leq \mu \leq \overline{X} + u_{1-\alpha_1/2} \frac{\sigma}{\sqrt{n}}, \right.$$
$$\left. \frac{nS^2}{\chi^2_{n-1;1-\alpha_2/2}} \leq \sigma^2 \leq \frac{nS^2}{\chi^2_{n-1;\alpha_2/2}} \right\}, \quad (16)$$

where $\overline{X}$ is the sample mean, $S^2 = (1/n) \sum_{i=1}^n (X_i - \overline{X})^2$ is the sample variance, $u_{1-\alpha_1/2}$ is the upper $\alpha_1/2$ percentile of a standard normal distribution, and $\chi^2_{n-1;\alpha_2/2}$ and $\chi^2_{n-1;1-\alpha_2/2}$ are the lower and upper $\alpha_2/2$ percentiles of a χ^2_{n-1} distribution. The shape of that region is illustrated in Figure 3. Values of α_1 and α_2 yielding a region of smallest possible size for a fixed confidence level are given in [3].

Let $\mathscr{P}$ denote the set of Gaussian distributions with parameters contained in confidence region $\mathscr{R}$. Applying the principle outlined in Section 3, we may obtain the q-MCD possibility distribution poss* for any x by maximizing poss$(x; \mu, \sigma)$ given by (15) with respect to μ and σ, under the constraint $(\mu, \sigma^2) \in \mathscr{R}$. The result is given by the following proposition.

Proposition 1. *The q-MCD possibility distribution poss* associated with the Mood confidence confidence region $\mathscr{R}$ at level $(1 - \alpha_1)(1 - \alpha_2)$ is*

$$poss^*(x) = \begin{cases} poss(x; \widehat{\mu^-}, \widehat{\sigma^+}) & \text{if } x < \widehat{\mu^-} \\ 1 & \text{if } \widehat{\mu^-} \leq x \leq \widehat{\mu^+} \\ poss(x; \widehat{\mu^+}, \widehat{\sigma^+}) & \text{if } x > \widehat{\mu^+}, \end{cases} \quad (17)$$

with

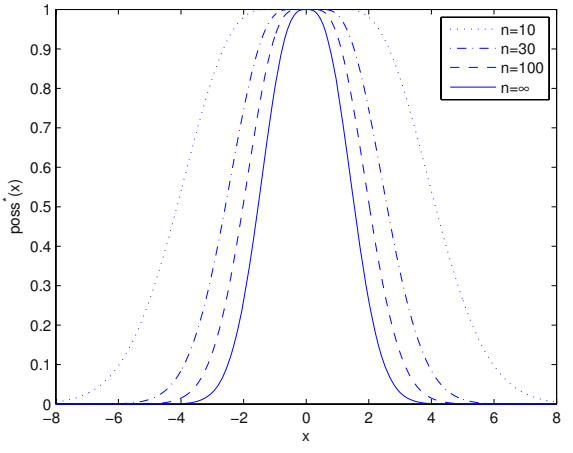


Fig. 4. Plot of $\text{poss}^*(x)$ for $\overline{x} = 0$, $s^2 = 1$, $\alpha = 0.1$, $\alpha_1 = \alpha_2$, and $n = 10, 30, 100$ and ∞

$$\widehat{\sigma}^+ = \left(\frac{nS^2}{\chi^2_{n-1;\alpha_2/2}} \right)^{1/2},$$

$$\widehat{\mu}^- = \overline{X} - u_{1-\alpha_1/2} \frac{\widehat{\sigma}^+}{\sqrt{n}}, \quad \widehat{\mu}^+ = \overline{X} + u_{1-\alpha_1/2} \frac{\widehat{\sigma}^+}{\sqrt{n}}.$$

Proof. We have by definition

$$\text{poss}^*(x) = \sup_{(\mu,\sigma^2)\in\mathscr{R}} \text{poss}(x;\mu,\sigma).$$

If $x \in [\widehat{\mu}^-, \widehat{\mu}^+]$, then we can get $\text{poss}(x,\mu,\sigma) = 1$ by setting $\mu = x$ and $\sigma = \widehat{\sigma}^+$. If $x < \widehat{\mu}^-$, then the value 1 cannot be reached. However, we obtain using standard calculus for $x < \mu$:

$$\frac{\partial \text{poss}(x;\mu,\sigma)}{\partial \mu} = -\frac{(x-\mu)^2}{\sigma^3\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) < 0$$

and

$$\frac{\partial \text{poss}(x;\mu,\sigma)}{\partial \sigma} = \frac{(\mu-x)^3}{\sigma^4\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) > 0.$$

Consequently, $\text{poss}(x;\mu,\sigma)$ is maximized by jointly minimizing μ and maximizing σ, and the maximum is reached for $(\mu,\sigma) = (\widehat{\mu}^-, \widehat{\sigma}^+)$. Similarly, we get for $x > \widehat{\mu}^+$:

$$\frac{\partial \text{poss}(x;\mu,\sigma)}{\partial \mu} = \frac{(x-\mu)^2}{\sigma^3\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) > 0$$

and

$$\frac{\partial \text{poss}(x;\mu,\sigma)}{\partial \sigma} = \frac{(x-\mu)^3}{\sigma^4\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) > 0.$$

Consequently, the maximum of $\text{poss}(x,\mu,\sigma)$ for $x > \widehat{\mu}^+$ is reached for $(\mu,\sigma) = (\widehat{\mu}^+, \widehat{\sigma}^+)$. □

Figure 4 shows the possibility distribution $\text{poss}^*(x)$ for $\overline{x} = 0$, $s^2 = 1$, $\alpha = 0.1$ and various values of n. The case $n = \infty$ corresponds to the situation where parameters μ and σ^2 are known: in that case, poss^* is simply que q-LC isopignistic possibility distribution induced by the normal pignistic distribution with $\mu = 0$ and $\sigma^2 = 1$.

5 Conclusion

A new method for generating a belief function from statistical data in the TBM framework has been presented. The starting point of this method is the assumption that, if the probability distribution $\mathbb{P}_X$ of a random variable is known, then the belief function quantifying our belief regarding a future realization of X should be such that its pignistic probability distribution equals $\mathbb{P}_X$. In the realistic situation where $\mathbb{P}_X$ is unknown but a random sample of X is available, it is possible to build a set $\mathscr{P}$ of probability distributions containing $\mathbb{P}$ with some confidence level. Following the LCP, it is then reasonable to impose that the sought belief function be q-less committed than all belief functions whose pignistic probability distribution is in $\mathscr{P}$. Our method selects the q-most committed consonant belief function verifying this property, referred to as the q-MCD possibility distribution induced by $\mathscr{P}$. This general principle has been illustrated in the case of the normal distribution.

In conjunction with the General Bayesian Theorem [16,6], the q-LC isopignistic transformation has proved useful to tackle classification problems using the TBM [13]. In this approach, the parameters of the pignistic distributions were assumed to be given by experts or estimated using large samples. Using the tools presented in this paper, it will be possible to apply this methodology to a wider range of problems where only small datasets are available. Future work in this direction will be reported in forthcoming papers.

References

1. Aregui, A., Denœux, T.: Constructing predictive belief functions from continuous sample data using confidence bands. In: De Cooman, G., Vejnarová, J., Zaffalon, M. (eds.) Proceedings of the Fifth International Symposium on Imprecise Probability: Theories and Applications (ISIPTA '07), Czech Republic, pp. 11–20 (July 2007)
2. Aregui, A., Denœux, T.: Fusion of one-class classifiers in the belief function framework. In: Proceedings of the 10th Int. Conf. on Information Fusion, Quebec, Canada (July 2007)
3. Arnold, B.C., Shavelle, R.M.: Joint confidence sets for the mean and variance of a normal distribution. The American Statistician 52(2), 133–140 (1998)
4. Dempster, A.P.: Upper and lower probabilities generated by a random closed interval. Annals of Mathematical Statistics 39(3), 957–966 (1968)
5. Denœux, T.: Constructing belief functions from sample data using multinomial confidence regions. International Journal of Approximate Reasoning 42(3), 228–252 (2006)
6. Denœux, T., Smets, P.: Classification using belief functions: the relationship between the case-based and model-based approaches. IEEE Transactions on Systems, Man and Cybernetics B 36(6), 1395–1406 (2006)
7. Dubois, D., Prade, H.: On several representations of an uncertainty body of evidence. In: Gupta, M.M., Sanchez, E. (eds.) Fuzzy Information and Decision Processes, North-Holland, New-York, pp. 167–181 (1982)

8. Dubois, D., Prade, H.: A set-theoretic view of belief functions: logical operations and approximations by fuzzy sets. International Journal of General Systems 12(3), 193–226 (1986)
9. Dubois, D., Prade, H., Smets, P.: New semantics for quantitative possibility theory. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, pp. 410–421. Springer, Heidelberg (2001)
10. Dubois, D., Prade, H., Smets, P.: A definition of subjective possibility. International Journal of Approximate Reasoning (2007) (in press)
11. Hacking, I.: Logic of Statistical Inference. Cambridge University Press, Cambridge (1965)
12. Masson, M.-H., Denœux, T.: Inferring a possibility distribution from empirical data. Fuzzy Sets and Systems 157(3), 319–340 (2006)
13. Ristic, B., Smets, P.: Target classification approach based on the belief function theory. IEEE Transactions on Aerospace and Electronic Systems 41(2), 574–583 (2005)
14. Shafer, G.: A mathematical theory of evidence. Princeton University Press, Princeton (1976)
15. Smets, P.: Un modèle mathématico-statistique simulant le processus du diagnostic médical. PhD thesis, Université Libre de Bruxelles, Brussels, Belgium (in French) (1978)
16. Smets, P.: Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem. International Journal of Approximate Reasoning 9, 1–35 (1993)
17. Smets, P.: Belief induced by the partial knowledge of the probabilities. In: D.H., et al. (eds.) Uncertainty in AI'94, pp. 523–530. Morgan Kaufmann, San Mateo (1994)
18. Smets, P.: Practical uses of belief functions. In: Laskey, K.B., Prade, H. (eds.) Uncertainty in Artificial Intelligence 15 (UAI99), Stockholm, Sweden, pp. 612–621 (1999)
19. Smets, P.: Belief functions on real numbers. International Journal of Approximate Reasoning 40(3), 181–223 (2005)
20. Smets, P.: Decision making in the TBM: the necessity of the pignistic transformation. International Journal of Approximate Reasoning 38, 133–147 (2005)
21. Smets, P., Kennes, R.: The Transferable Belief Model. Artificial Intelligence 66, 191–243 (1994)
22. Yager, R.R.: The entailment principle for Dempster-Shafer granules. Int. J. of Intelligent Systems 1, 247–262 (1986)

On the Orthogonal Projection of a Belief Function

Fabio Cuzzolin

INRIA Rhone-Alpes
Fabio.Cuzzolin@inrialpes.fr

Abstract. In this paper we study a new probability associated with any given belief function b, i.e. the orthogonal projection $\pi[b]$ of b onto the probability simplex $\mathcal{P}$. We provide an interpretation of $\pi[b]$ in terms of a redistribution process in which the mass of each focal element is equally distributed among its subsets, establishing an interesting analogy with the pignistic transformation. We prove that orthogonal projection commutes with convex combination just as the pignistic function does, unveiling a decomposition of $\pi[b]$ as convex combination of basis pignistic functions. Finally we discuss the norm of the difference between orthogonal projection and pignistic function in the case study of a quaternary frame, as a first step towards a more comprehensive picture of their relation.

1 Introduction

The theory of evidence (ToE) is one of the most popular uncertainty theories, thanks perhaps to its nature of quite natural extension of the classical Bayesian methodology. Indeed, the notion of *belief function* (b.f.) [1] generalizes that of finite probability, with classical probabilities forming a subclass $\mathcal{P}$ of b.f.s called *Bayesian* belief functions. The interplay of belief and Bayesian functions is of course of great interest in the theory of evidence. In particular, many people worked on the problem of finding a probabilistic approximation of an arbitrary belief function. Several papers [2,3,4,5,6,7,8,9,10] have been published on this issue, mainly in order to find efficient implementations of the rule of combination aiming to reduce the number of focal elements. The connection between belief functions and probabilities is as well crucial in Smets' "Transferable Belief Model" [11].

The study of the links between belief functions and probabilities has recently been posed in a geometric setup [12,13]. In robust Bayesian statistics, there is a large literature on the study of convex sets of probability distributions [14,15,16,17]. On our side, in a series of works [18,19] we proposed a geometric interpretation of the theory of evidence in which belief functions are represented as points of a simplex called *belief space* $\mathcal{B}$, a polytope whose vertices are all the b.f.s focused on a single event A, $m_b(A) = 1$, $m_b(B) = 0\ \forall B \neq A$. The region $\mathcal{P}$ of Bayesian b.f.s is also a simplex, part of the border of $\mathcal{B}$. The relation between belief and probability measures can then be naturally studied in this framework.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 356–367, 2007.

In this paper we use tools provided by the geometric approach to introduce a new probability function $\pi[b]$ associated with any given belief function b, precisely the *orthogonal projection* of b onto the probability simplex $\mathcal{P}$. We thoroughly discuss its interpretation and properties, and its relations with other known Bayesian approximations of belief functions, i.e. pignistic function and relative plausibility of singletons. We show that $\pi[b]$ is inherently related to a redistribution process similar to that of the pignistic transformation, in which though the mass of each focal element is reassigned to *all its subsets*. We prove that, just as the pignistic function does, the orthogonal projection *commutes* with respect to the convex combination operator, yielding an interesting decomposition of $\pi[b]$ in terms of convex combination of basis pignistic functions.

2 A Geometric Approach to the ToE

A *basic belief assignment* (b.b.a.) over a finite set or "frame of discernment" [1] Θ is a function $m : 2^\Theta \to [0,1]$ on its power set $2^\Theta \doteq \{A \subseteq \Theta\}$ such that $m(\emptyset) = 0$, $\sum_{A\subseteq\Theta} m(A) = 1$, $m(A) \geq 0 \ \forall A \subseteq \Theta$. Subsets of Θ associated with non-zero values of m are called *focal elements*. The *belief function* $b : 2^\Theta \to [0,1]$ associated with a basic belief assignment m on Θ is defined as: $b(A) = \sum_{B\subseteq A} m(B)$. The unique b.b.a. m_b associated with a given b.f. b can be recovered by means of the Moebius inversion formula $m_b(A) = \sum_{B\subseteq A}(-1)^{|A-B|}b(B)$. In the ToE a probability function is simply a special belief function assigning non-zero masses to singletons only (*Bayesian* b.f.): $m_b(A) = 0$, $|A| > 1$. A dual mathematical representation of the evidence encoded by a b.f. b is the *plausibility function* (pl.f.) $pl_b : 2^\Theta \to [0,1]$, where the plausibility $pl_b(A)$ of an event A is given by $pl_b(A) \doteq 1 - b(A^c) = 1 - \sum_{B\subseteq A^c} m_b(B) = \sum_{B\cap A\neq\emptyset} m_b(B)$ where A^c denotes the complement of A in Θ.

Motivated by the search for a meaningful probabilistic approximation of belief functions we introduced the notion of *belief space* [19], as the space of all belief functions defined on a given frame of discernment Θ. A belief function $b : 2^\Theta \to [0,1]$ is completely specified by its $N-1$ belief values $\{b(A), A \subseteq \Theta, A \neq \emptyset\}$, $N \doteq 2^{|\Theta|}$, and can then be represented as a point of $\mathbb{R}^{N-1}$. The belief space associated with Θ is the set of points $\mathcal{B}_\Theta$ of $\mathbb{R}^{N-1}$ corresponding to a belief function. We will assume the domain Θ fixed, and denote the belief space with $\mathcal{B}$. It is not difficult to prove [19] that $\mathcal{B}$ is convex. Let $b_A \doteq b \in \mathcal{B}$ s.t. $m_b(A) = 1$, $m_b(B) = 0 \ \forall B \neq A$ be the unique belief function assigning all the mass to a single subset A of Θ (A-th *basis belief function*). It can be proved that [19] the belief space $\mathcal{B}$ is the convex closure of all basis belief functions b_A: $\mathcal{B} = Cl(b_A, \emptyset \subsetneq A \subseteq \Theta)$, where Cl denotes the convex closure operator:

$$Cl(b_1, ..., b_k) = \Big\{ b \in \mathcal{B} : b = \alpha_1 b_1 + \cdots + \alpha_k b_k : \textstyle\sum_i \alpha_i = 1, \alpha_i \geq 0 \ \forall i \Big\}. \quad (1)$$

The convex space delimited by a collection of (affinely independent [20]) points is called *simplex*. Each b.f. $b \in \mathcal{B}$ can be written as a convex sum as $b = \sum_{\emptyset\subsetneq A\subseteq\Theta} m_b(A) b_A$. Geometrically, the b.b.a. m_b is the set of coordinates of b

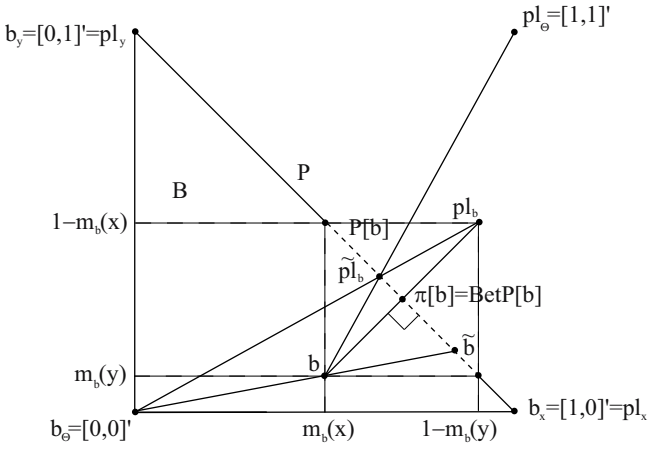


Fig. 1. In a binary frame $\Theta_2 = \{x, y\}$ the belief space $\mathcal{B}$ is a simplex with vertices $\{b_\Theta = [0,0]', b_x = [1,0]', b_y = [0,1]'\}$. A belief function b and the corresponding plausibility function pl_b are always located in symmetric positions with respect to the set $\mathcal{P}$ of probabilities on Θ. The associated relative plausibility $\tilde{pl}_b$ and belief $\tilde{b}$ of singletons are shown as the intersections of the probabilistic subspace with the line joining pl_b and $b_\Theta = [0,0]'$ and the line passing through b and b_Θ respectively. The other Bayesian functions related to b all coincide with the center of the segment of consistent probabilities $\mathcal{P}[b]$.

in the simplex $\mathcal{B}$. Analogously, the set $\mathcal{P}$ of all Bayesian belief functions on Θ is the simplex determined by all the basis b.f.s associated with singletons: $\mathcal{P} = Cl(b_x, x \in \Theta)$. PL.F.s can also be seen as points of $\mathbb{R}^{N-1}$ [19].

3 Orthogonal Projection: Binary Case

It may be helpful to visually render these concepts in a simple example. Figure 1 shows the geometry of belief and plausibility functions for a binary frame $\Theta_2 = \{x, y\}$. Since $b(\Theta) = pl_b(\Theta) = 1$ for all b, we can represent belief and plausibility vectors as points of a plane with coordinates $b = [b(x) = m_b(x), b(y) = m_b(y)]'$, $pl_b = [pl_b(x) = 1 - m_b(y), pl_b(y) = 1 - m_b(x)]'$ respectively.

Each pair of functions (b, pl_b) determines a line which is *orthogonal* to $\mathcal{P}$, where b and pl_b lie on symmetric positions on the two sides of the Bayesian region. The set of probabilities $\mathcal{P}[b]$ consistent with b, i.e. $\mathcal{P}[b] \doteq \{p \in \mathcal{P} : p(A) \geq b(A)\ \forall A \subseteq \Theta\}$ in the simple binary case forms a segment in $\mathcal{P}$ (see Figure 1 again), whose center of mass is well known [21,22,18] to be Smets' *pignistic function* [23]

$$BetP[b] = \sum_{x \in \Theta} b_x \sum_{A \supseteq \{x\}} \frac{m_b(A)}{|A|} = b_x\Big(m_b(x) + \frac{m_b(\Theta)}{2}\Big) + b_y\Big(m_b(y) + \frac{m_b(\Theta)}{2}\Big). \tag{2}$$

It is interesting to notice though that it also coincides with the *orthogonal projection* $\pi[b]$ of b onto $\mathcal{P}$: $\pi[b] = BetP[b] = \overline{\mathcal{P}}[b]$. On the other side, both *relative plausibility* $\tilde{pl}_b$ and *relative belief* $\tilde{b}$ *of singletons*

$$\tilde{pl}_b(x) \doteq \frac{pl_b(x)}{\sum_{y\in\Theta} pl_b(y)}, \;\; \tilde{b}(x) \doteq \frac{b(x)}{\sum_{y\in\Theta} b(y)} \tag{3}$$

even though consistent with b, do not follow the same scheme.

In the following we will study the geometry of the orthogonal complement of $\mathcal{P}$ and analyze the properties of the associated Bayesian function, the orthogonal projection $\pi[b]$ of b onto the probability simplex $\mathcal{P}$.

4 Orthogonal Projection

Let us then denote with $a(v_1, .., v_k)$ the affine subspace of some Cartesian space $\mathbb{R}^m$ generated by the points $v_1, ..., v_k \in \mathbb{R}^m$, i.e. the set $\{v \in \mathbb{R}^m : v = \alpha_1 v_1 + \cdots + \alpha_k v_k, \sum_i \alpha_i = 1\}$. The orthogonal projection $\pi[b]$ of b onto $a(\mathcal{P})$ is obviously guaranteed to exist as $a(\mathcal{P})$ is nothing but a linear subspace of $\mathbb{R}^{N-1}$ (on which b lives). By definition, the orthogonal projection $\pi[b]$ is the solution of the optimization problem

$$\pi[b] = \arg\min_{p\in\mathcal{P}} \|p - b\|_{L_2} = \arg\min_{p\in\mathcal{P}} \sqrt{\sum_{A\subseteq\Theta} |p(A) - b(A)|^2} \tag{4}$$

and is then naturally the unique solution of the Bayesian approximation problem when choosing the L_2 distance in the belief space.

An explicit calculation of $\pi[b]$ requires a description of the orthogonal complement of $a(\mathcal{P})$ in $\mathbb{R}^{N-1}$. Let $n = |\Theta|$ be the cardinality of Θ.

4.1 General Form of the Orthogonal Projection

We need to find a necessary and sufficient condition for an arbitrary vector $v = \sum_{A\subseteq\Theta} v_A X_A$, where $\{X_A, A \subseteq \Theta\}$ is a reference frame in $\mathbb{R}^{N-1}$, to be orthogonal to the probabilistic subspace $a(\mathcal{P})$. If we compute the scalar product $\langle v, b_y - b_x\rangle$ between v and the generators $b_y - b_x$ of $a(P)$ we get

$$\Big\langle \sum_{A\subseteq\Theta} v_A X_A, b_y - b_x \Big\rangle = \sum_{A\subseteq\Theta} v_A [b_y - b_x](A).$$

After remembering that, by definition, $b_A(B) = 1$ if $B \supseteq A$, 0 elsewhere, we can see that these vectors display a special symmetry

$$b_y - b_x(A) = \begin{cases} 1 & A \supseteq \{y\}, A \not\supset \{x\} \\ 0 & A \supseteq \{x\}, \{y\} \; or \; A \not\supset \{x\}, \{y\} \\ -1 & A \not\supset \{y\}, A \supseteq \{x\} \end{cases}$$

which allows us to write $\langle v, b_y - b_x \rangle = \sum_{A \supseteq y, A \not\supset \{x\}} v_A - \sum_{A \supseteq \{x\}, A \not\supset \{y\}} v_A$. The orthogonal complement $a(\mathcal{P})^\perp$ of $a(\mathcal{P})$ will then be expressed as

$$a(\mathcal{P})^\perp = \Big\{ v : \sum_{A \supseteq \{y\}, A \not\supset \{x\}} v_A = \sum_{A \supseteq \{x\}, A \not\supset \{y\}} v_A \ \forall y \neq x \Big\}.$$

If the vector v, in particular, is a belief function

$$\sum_{A \supseteq \{y\}, A \not\supset \{x\}} b(A) = \sum_{A \supseteq \{y\}, A \not\supset \{x\}} \sum_{B \subseteq A} m_b(B) = \sum_{B \subseteq \{x\}^c} m_b(B) \cdot 2^{n-1-|B \cup \{y\}|}$$

since $2^{n-1-|B\cup\{y\}|}$ is the number of subsets A of $\{x\}^c$ containing both B and y, and the orthogonality condition becomes

$$\sum_{B \subseteq \{x\}^c} m_b(B) 2^{1-|B\cup\{y\}|} = \sum_{B \subseteq \{y\}^c} m_b(B) 2^{1-|B\cup\{x\}|}$$

$\forall y \neq x$, after erasing the common factor 2^{n-2}. Now, events $B \subseteq \{x, y\}^c$ appear in both summations, with the same coefficient (since $|B \cup \{x\}| = |B \cup \{y\}| = |B|+1$) and the equation reduces to

$$\sum_{B \supseteq \{y\}, B \not\supset \{x\}} m_b(B) 2^{1-|B|} = \sum_{B \supseteq \{x\}, B \not\supset \{y\}} m_b(B) 2^{1-|B|} \tag{5}$$

$\forall y \neq x$, the desired orthogonality condition. (5) can be used to prove that [24]

Theorem 1. *The orthogonal projection $\pi[b]$ of b onto $a(\mathcal{P})$ can be expressed in terms of the b.b.a. m_b of b in the following equivalent forms:*

$$\sum_{A \supseteq \{x\}} m_b(A) 2^{1-|A|} + \sum_{A \subseteq \Theta} m_b(A) \Big(\frac{1 - |A| 2^{1-|A|}}{n} \Big) \tag{6}$$

$$\sum_{A \supseteq \{x\}} m_b(A) \Big(\frac{1 + |A^c| 2^{1-|A|}}{n} \Big) + \sum_{A \not\supset \{x\}} m_b(A) \Big(\frac{1 - |A| 2^{1-|A|}}{n} \Big). \tag{7}$$

From (7) we can see that $\pi[b]$ *is indeed a probability*, since both $1 + |A^c| 2^{1-|A|} \geq 0$ and $1 - |A| 2^{1-|A|} \geq 0 \ \forall |A| = 1, ..., n$. This is not at all trivial, as $\pi[b]$ is the projection of b onto the *affine* space $a(\mathcal{P})$, and could have in principle assigned negative masses to one or more singletons. $\pi[b]$ is hence a valid candidate to the role of probabilistic approximation of the b.f. b.

Unnormalized case. It is interesting to note that the above results hold for *unnormalized* belief functions [25] too. The orthogonality results of Section 4.1 are still valid as the proof of Theorem 1 [24] does not concern the mass of the empty set. The orthogonal projection $\pi[b]$ of a u.b.f. b is then well defined and

is still given by Equations (6),(7) where this time the summations on the right hand side include $\emptyset$ too:

$$\pi[b](x) = \sum_{A \supseteq \{x\}} m_b(A) 2^{1-|A|} + \sum_{\emptyset \subseteq A \subseteq \Theta} m_b(A) \Big(\frac{1 - |A| 2^{1-|A|}}{n} \Big)$$

$$\pi[b](x) = \sum_{A \supseteq \{x\}} m_b(A) \Big(\frac{1 + |A^c| 2^{1-|A|}}{n} \Big) + \sum_{\emptyset \subseteq A \not\supseteq \{x\}} m_b(A) \Big(\frac{1 - |A| 2^{1-|A|}}{n} \Big).$$

4.2 Orthogonality Flag and Redistribution Process

Theorem 1 does not apparently provide any intuition about the meaning of $\pi[b]$ in terms of degrees of belief. If we process Equation (7) though we can reduce π to a new Bayesian function strictly related to the pignistic function [24].

Theorem 2. *$\pi[b] = \bar{\mathcal{P}}(1 - k_O[b]) + k_O[b] O[b]$, where $\bar{\mathcal{P}}$ is the uniform probability and $O[b](x)$ is the Bayesian b.f.*

$$O[b](x) = \frac{\bar{O}[b](x)}{k_O[b]} = \frac{\sum_{A \supseteq \{x\}} m_b(A) 2^{1-|A|}}{\sum_{A \subseteq \Theta} m_b(A) |A| 2^{1-|A|}}. \tag{8}$$

As $0 \leq |A| 2^{1-|A|} \leq 1$ for all $A \subseteq \Theta$, $k_O[b]$ assumes values in the interval $[0, 1]$. Theorem 2 then implies that the orthogonal projection is always located on the line segment joining the uniform, non-informative probability $\bar{\mathcal{P}}$ and the Bayesian b.f. $O[b]$. By Equation (8) it turns out that $\pi[b] = \bar{\mathcal{P}}$ iff $O[b] = \bar{\mathcal{P}}$ (since $k_O[b] > 0$). The meaning to attribute to $O[b]$ becomes clear when we notice that the condition (5) under which a b.f. b is orthogonal to $a(\mathcal{P})$ can be rewritten as $\sum_{B \supseteq \{y\}, B \not\supseteq \{x\}} m_b(B) 2^{1-|B|} + \sum_{B \supseteq \{y\}, \{x\}} m_b(B) 2^{1-|B|} = \sum_{B \supseteq \{x\}, B \not\supseteq \{y\}} m_b(B) 2^{1-|B|} + \sum_{B \supseteq \{y\}, \{x\}} m_b(B) 2^{1-|B|} \equiv \sum_{B \supseteq \{y\}} m_b(B) 2^{1-|B|} = \sum_{B \supseteq \{x\}} m_b(B) 2^{1-|B|} \equiv \bar{O}[b](x) = const \equiv O[b](x) = const = \bar{\mathcal{P}} \; \forall \; x \in \Theta$. Therefore $\pi[b] = \bar{\mathcal{P}}$ iff $b \perp a(\mathcal{P})$, and $O - \bar{\mathcal{P}}$ measures the non-orthogonality of b with respect to $\mathcal{P}$. $O[b]$ deserves then the name of *orthogonality flag*.

A compelling link can be drawn between orthogonal projection and pignistic function through the orthogonality flag $O[b]$. Let us define the two b.f.

$$b_{||} \doteq \frac{1}{k_{||}} \sum_{A \subseteq \Theta} \frac{m_b(A)}{|A|} b_A, \qquad b_{2^{||}} \doteq \frac{1}{k_{2^{||}}} \sum_{A \subseteq \Theta} \frac{m_b(A)}{2^{|A|}} b_A,$$

$k_{||}$ and $k_{2^{||}}$ their normalization factors.

Theorem 3. *$O[b]$ is the relative plausibility of singletons of $b_{2^{||}}$; $BetP[b]$ is the relative plausibility of singletons of $b_{||}$.*

Proof. By definition of plausibility function

$$pl_{b_{2^{||}}}(x) = \sum_{A \supseteq \{x\}} m_{b_{2^{||}}}(A) = \frac{1}{k_{2^{||}}} \sum_{A \supseteq \{x\}} \frac{m_b(A)}{2^{|A|}} = \frac{\bar{O}[b]}{2 k_{2^{||}}},$$

$$\sum_{x \in \Theta} pl_{b_{2^{||}}}(x) = \frac{1}{k_{2^{||}}} \sum_{x \in \Theta} \sum_{A \supseteq \{x\}} \frac{m_b(A)}{2^{|A|}} = \frac{k_O[b]}{2 k_{2^{||}}}$$

as $k_O[b]$ is the normalization factor for $\bar{O}[b]$:

$$\sum_{x\in\Theta}\bar{O}[b](x)=\sum_{x\in\Theta}\sum_{A\supseteq\{x\}}m_b(A)2^{1-|A|}=\sum_{A\subseteq\Theta}m_b(A)|A|2^{1-|A|}=k_O[b].$$

Hence $\tilde{pl}_{b_{2^{||}}}(x)=\bar{O}[b]/k_O[b]=O[b]$, i.e.

$$pl_{b_{||}}(x)=\sum_{A\supseteq\{x\}}m_{b_{||}}(A)=\frac{1}{k_{||}}\sum_{A\supseteq\{x\}}\frac{m_b(A)}{|A|}=\frac{1}{k_{||}}BetP[b](x)$$

and since $\sum_x BetP[b](x)=1$, $\tilde{pl}_{b_{||}}(x)=BetP[b](x)$.

The two functions $b_{||}$ and $b_{2^{||}}$ represent two different processes acting on b (see Figure 2). The first one redistributes the mass of each focal element among its *singletons* (yielding directly a Bayesian b.f. $BetP[b]$). The second one distributes the b.b.a. of each event A among its *subsets* $B\subseteq A$ ($\emptyset$, A included). In this second case we get an *unnormalized* [25] b.f. b^U with $m_{b^U}(A)=\sum_{B\supseteq A}\frac{m_b(B)}{2^{|B|}}$ whose relative belief of singletons (3) $\tilde{b^U}$ is in fact the orthogonality flag $O[b]$.

Example. Let us consider as an example the belief function b on the ternary frame: $m_b(x)=0.1$, $m_b(y)=0$, $m_b(z)=0.2$, $m_b(\{x,y\})=0.3$, $m_b(\{x,z\})=0.1$, $m_b(\{y,z\})=0$, $m_b(\Theta)=0.3$. To get the orthogonality flag $O[b]$ we need to apply the redistribution process of Figure 2 to each focal element of b. In this case their masses are divided among their subsets as follows:

$$\begin{array}{ll}
m_b(x)=0.1 & \mapsto m'(x)=m'(\emptyset)=0.1/2=0.05,\\
m_b(z)=0.2 & \mapsto m'(z)=m'(\emptyset)=0.2/2=0.1,\\
m_b(\{x,y\})=0.3 & \mapsto m'(\{x,y\})=m'(x)=m'(y)=m'(\emptyset)=0.3/4=0.075,\\
m_b(\{x,z\})=0.1 & \mapsto m'(\{x,z\})=m'(x)=m'(z)=m'(\emptyset)=0.1/4=0.025,\\
m_b(\Theta)=0.3 & \mapsto m'(\Theta)=m'(\{x,y\})=m'(\{x,z\})=m'(\{y,z\})=\\
 & \quad =m'(x)=m'(y)=m'(z)=m'(\emptyset)=0.3/8=0.0375.
\end{array}$$

By summing contributions related to singletons on the right hand side we get

$$m_{b^U}(x)=0.05+0.075+0.025+0.0375=0.1875,$$
$$m_{b^U}(y)=0.075+0.0375=0.1125,\quad m_{b^U}(z)=0.1+0.025+0.0375=0.1625$$

whose sum is the normalization factor $k_O[b]=m_{b^U}(x)+m_{b^U}(y)+m_{b^U}(z)=0.4625$ and by normalizing $O[b]=[0.405,0.243,0.351]'$. The orthogonal projection $\pi[b]$ is finally the convex combination of $O[b]$ and $\bar{\mathcal{P}}=[1/3,1/3,1/3]'$ with coordinate $k_O[b]$: $\pi[b]=\bar{\mathcal{P}}(1-k_O[b])+k_O[b]O[b]=[1/3,1/3,1/3]'\cdot(1-0.4625)+0.4625\cdot[0.405,0.243,0.351]'=[0.366,0.291,0.342]'$.

4.3 Orthogonal Projection and Convex Combination

As a confirmation of this relationship, orthogonal projection and pignistic function both commute with convex combination.

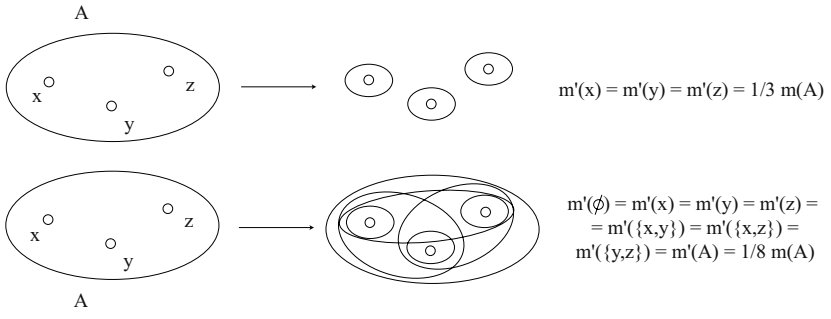


Fig. 2. Redistribution processes associated with pignistic transformation and orthogonal projection. In the pignistic transformation (top) the mass of each focal element is distributed among its elements. In the orthogonal projection (bottom), instead (through the orthogonality flag), the mass of each f.e. is divided among its subsets. In both cases, the related relative belief of singletons yields a Bayesian belief function.

Theorem 4. *Orthogonal projection and convex combination commute, i.e. if* $\alpha_1 + \alpha_2 = 1$ *then* $\pi[\alpha_1 b_1 + \alpha_2 b_2] = \alpha_1 \pi[b_1] + \alpha_2 \pi[b_2]$.

Proof. By Theorem 2 $\pi[b] = (1 - k_O[b])\bar{\mathcal{P}} + \bar{O}[b]$ where the coefficient is $k_O[b] = \sum_{A \subseteq \Theta} m_b(A)|A|2^{1-|A|}$ and $\bar{O}[b](x) = \sum_{A \supseteq \{x\}} m_b(A) 2^{1-|A|}$. Hence

$$k_O[\alpha_1 b_1 + \alpha_2 b_2] = \sum_{A \subseteq \Theta} \big(\alpha_1 m_{b_1}(A) + \alpha_2 m_{b_2}(A)\big)|A|2^{1-|A|} = \alpha_1 k_O[b_1] + \alpha_2 k_O[b_2],$$
$$\bar{O}[\alpha_1 b_1 + \alpha_2 b_2](x) = \sum_{A \supseteq \{x\}} \big(\alpha_1 m_{b_1}(A) + \alpha_2 m_{b_2}(A)\big) 2^{1-|A|} = \alpha_1 \bar{O}[b_1] + \alpha_2 \bar{O}[b_2]$$

which in turn implies (since $\alpha_1 + \alpha_2 = 1$) $\pi[\alpha_1 b_1 + \alpha_2 b_2] = (1 - \alpha_1 k_O[b_1] - \alpha_2 k_O[b_2])\bar{\mathcal{P}} + \alpha_1 \bar{O}[b_1] + \alpha_2 \bar{O}[b_2] = \alpha_1 \big[(1 - k_O[b_1])\bar{\mathcal{P}} + \bar{O}[b_1]\big] + \alpha_2 \big[(1 - k_O[b_2])\bar{\mathcal{P}} + \bar{O}[b_2]\big] = \alpha_1 \pi[b_1] + \alpha_2 \pi[b_2]$.

This property can be used to find an alternative expression of the orthogonal projection as convex combination of the pignistic functions associated with all basis belief functions.

Lemma 1. *The orthogonal projection of a basis belief function* b_A *is given by*

$$\pi[b_A] = (1 - |A|2^{1-|A|})\bar{\mathcal{P}} + |A|2^{1-|A|}\bar{\mathcal{P}}_A,$$

with $\bar{\mathcal{P}}_A = \frac{1}{|A|} \sum_{x \in A} b_x$ *the center of mass of all probabilities with support in* A.

Proof. By Equation (8) $k_O[b_A] = |A|2^{1-|A|}$, so that $\bar{O}[b_A](x) = 2^{1-|A|}$ if $x \in A$, 0 otherwise. This implies

$$O[b_A](x) = \begin{cases} \frac{1}{|A|} & x \in A \\ 0 & x \notin A \end{cases} = \frac{1}{|A|} \sum_{x \in A} b_x = \bar{\mathcal{P}}_A.$$

Theorem 5. *The orthogonal projection can be expressed as a convex combination of all the non-informative probabilities with support on a single event A:*

$$\pi[b] = \bar{\mathcal{P}}\Big(1 - \sum_{A \neq \Theta} \alpha_A\Big) + \sum_{A \neq \Theta} \alpha_A \bar{\mathcal{P}}_A, \qquad \alpha_A \doteq m_b(A)|A|2^{1-|A|}. \tag{9}$$

Proof. $\pi[b] = \pi\Big[\sum_{A \subseteq \Theta} m_b(A) b_A\Big] = \sum_{A \subseteq \Theta} m_b(A)\pi[b_A]$ by Theorem 4, which by Lemma 1 becomes $\sum_{A \subseteq \Theta} m_b(A)[(1 - |A|2^{1-|A|})\bar{\mathcal{P}} + |A|2^{1-|A|}\bar{\mathcal{P}}_A] = (1 - \sum_{A \subseteq \Theta} m_b(A)|A|2^{1-|A|})\bar{\mathcal{P}} + \sum_{A \subseteq \Theta} m_b(A)|A|2^{1-|A|}\bar{\mathcal{P}}_A$ which can be written as

$$\Big(1 - \sum_{A \subseteq \Theta} m_b(A)|A|2^{1-|A|}\Big)\bar{\mathcal{P}} + \sum_{A \neq \Theta} m_b(A)|A|2^{1-|A|}\bar{\mathcal{P}}_A + m_b(\Theta)|\Theta|2^{1-|\Theta|}\bar{\mathcal{P}}$$

i.e. Equation (9).

As $\bar{\mathcal{P}}_A = BetP[b_A]$, we recognize that

$$BetP[b] = \sum_{A \subseteq \Theta} m_b(A) BetP[b_A],$$
$$\pi[b] = \sum_{A \neq \Theta} \alpha_A BetP[b_A] + \Big(1 - \sum_{A \neq \Theta} \alpha_A\Big) BetP[b_\Theta],$$

i.e. both orthogonal projection and pignistic function are convex combinations of all the basis pignistic functions. However, as $k_O[b_A] = |A|2^{1-|A|} < 1$ for $|A| > 2$, the orthogonal projection turns out to be closer to the vertices associated with events of lower cardinality (see Figure 3-left).

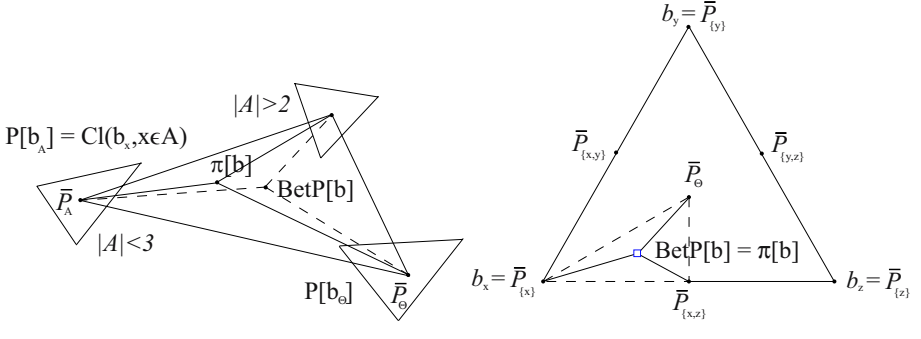


Fig. 3. Left: Orthogonal projection $\pi[b]$ and pignistic function $BetP[b]$ are both located on the simplex whose vertices are all the basis pignistic functions, i.e. the uniform probabilities associated with each single event A. However, the convex coordinates of $\pi[b]$ are weighted by a factor $k_O[b_A] = |A|2^{1-|A|}$, yielding a point which is closer to vertices related to lower size events. Right: Orthogonal projection and pignistic function coincide in the ternary case $\Theta_3 = \{x, y, z\}$.

Example: Ternary Case. Let us consider as an example a ternary frame $\Theta_3 = \{x, y, x\}$, and a belief function on Θ_3 with b.b.a. $m_b(x) = 1/3$, $m_b(\{x, z\}) = 1/3$, $m_b(\Theta_3) = 1/3$, $m_b(A) = 0$ $A \neq \{x\}, \{x, z\}, \Theta_3$. According to Equation (9)

$$\pi[b] = 1/3\bar{\mathcal{P}}_{\{x\}} + 1/3\bar{\mathcal{P}}_{\{x,z\}} + (1 - 1/3 - 1/3)\bar{\mathcal{P}} = \frac{1}{3}b_x + \frac{1}{3}\frac{b_x+b_z}{2} + \\ + \frac{1}{3}\frac{b_x+b_y+b_z}{3} = b_x(\frac{1}{3} + \frac{1}{6} + \frac{1}{9}) + b_z(\frac{1}{6} + \frac{1}{9}) + b_y\frac{1}{9} = \frac{11}{18}b_x + \frac{1}{9}b_y + \frac{5}{18}b_z$$

and the orthogonal projection is the barycenter of the simplex $Cl(\bar{\mathcal{P}}_{\{x\}}, \bar{\mathcal{P}}_{\{x,z\}}, \bar{\mathcal{P}})$ (see Figure 3-right). On the other side $BetP[b](x) = m_b(x) + \frac{m_b(x,z)}{2} + \frac{m_b(\Theta_3)}{3} = \frac{11}{18}$, $BetP[b](y) = \frac{1}{9}$, $BetP[b](z) = \frac{1}{6} + \frac{1}{9} = \frac{5}{18}$ i.e. $BetP[b] = \pi[b]$. This is true for each belief function $b \in \mathcal{B}_3$, since for the above expressions when $|\Theta| = 3$ $\alpha_A = m_b(A)$ for $|A| \leq 2$, and $1 - \sum_A \alpha_A = 1 - \sum_{A \neq \Theta} m_b(A) = m_b(\Theta)$.

4.4 A Quantitative Analysis of the Distance Between *BetP* and π

An exhaustive description of the relationship between orthogonal projection and pignistic function would require a quantitative analysis of their distance as the degrees of belief of b vary in the belief space.

Considered the fact that $\pi[b]$ is the solution of the Bayesian approximation problem when using the L_2 norm (4), a sensible choice is measuring their distance by computing the L_2 norm of their difference vector:

$$\|\pi[b] - BetP[b]\|_{L_2} \doteq \sqrt{\sum_{x \in \Theta} |\pi[b](x) - BetP[b](x)|^2}.$$

Let us then measure their difference in the simplest case in which they are distinct: a frame $\Theta = \{x, y, z, w\}$ of size 4. Their analytic expressions

$$\begin{aligned} BetP[b](x) &= \frac{1}{4}m_b(\Theta) + m_b(x) + \frac{1}{2}(m_b(\{x, y\}) + m_b(\{x, z\}) + m_b(\{x, w\})) + \\ &+ \frac{1}{3}(m_b(\{x, y, z\}) + m_b(\{x, y, w\}) + m_b(\{x, z, w\})); \\ \pi[b](x) &= \frac{1}{4}m_b(\Theta) + m_b(x) + \frac{1}{2}(m_b(\{x, y\}) + m_b(\{x, z\}) + m_b(\{x, w\})) + \\ &+ \frac{5}{16}(m_b(\{x, y, z\}) + m_b(\{x, y, w\}) + m_b(\{x, z, w\})) + \frac{1}{16}m_b(\{y, z, w\}) \end{aligned} \tag{10}$$

are very similar to each other. Basically the difference is that $\pi[b]$ counts also the masses of focal elements in $\{x\}^c$ (with a small contribution), while $BetP[b]$ by definition does not. If we compute their difference $BetP[b](x) - \pi[b](x) = \frac{1}{48}[m_b(\{x, y, z\}) + m_b(\{x, y, w\}) + m_b(\{x, z, w\}) - 3\ m_b(\{y, z, w\})]$ we can analyze the behavior of their L_2 distance as b varies. After introducing the simpler notation

$$y_1 = m_b(\{x, y, z\}), y_2 = m_b(\{x, y, w\}), y_3 = m_b(\{x, z, w\}), y_4 = m_b(\{y, z, w\}),$$

we can maximize (minimize) the norm above $(y_1 + y_2 + y_3 - 3y_4)^2 + (y_1 + y_2 + y_4 - 3y_3)^2 + (y_1 + y_3 + y_4 - 3y_2)^2 + (y_2 + y_3 + y_4 - 3y_1)^2$ by imposing

$\frac{\partial}{\partial y_i}\|BetP[b](\mathbf{y}) - \pi[b](\mathbf{y})\|^2 = 0$ subject to $y_1 + y_2 + y_3 + y_4 = 1$. The unique solution turns out to be $\mathbf{y} = [1/4, 1/4, 1/4, 1/4]'$ which corresponds to (after replacing this solution in (10) $BetP[b] = \pi[b] = \bar{\mathcal{P}}$ where $\bar{\mathcal{P}} = [1/4, 1/4, 1/4, 1/4]'$ is the uniform probability on Θ. The distance between pignistic function and orthogonal projection is minimal (zero) when all size 3 subsets have the same mass.

It is then natural to suppose that their difference must be maximal when all the mass is concentrated on a single size-3 event. This is in fact correct: $\|BetP[b] - \pi[b]\|^2$ is maximal and equal to $1^2 + 1^2 + 1^2 + (-3)^2 = 12$ when $y_i = 1$, $y_j = 0\ \forall j \neq i$, i.e. the mass of one among $\{x, y, z\}, \{x, y, w\}, \{x, z, w\}, \{y, z, w\}$ is one.

Other distances could of course be chosen to assess the difference between Bayesian approximations in the probability simplex: A natural generalization of L_2 is the Mahalanobis distance $\sqrt{(p - p')'\Sigma(p - p')}$ (where Σ is a covariance matrix) which is often used in statistics. Our intuition on the problem suggests that the above results should hold for a wide class of such functions: Experimental validation is though needed.

5 Conclusions

In this paper we introduced a new Bayesian b.f. associated with any given belief function b, i.e. the orthogonal projection of b onto the probability simplex $\mathcal{P}$, by definition the solution of the probabilistic approximation problem when using the classical L_2 distance. Even though $\pi[b]$ has been derived through purely geometric considerations, it exhibits strong links with the pignistic function. Its interpretation in terms of rationality principles similar to those formulated for the pignistic transformation is still unclear, as it is to decide whether or not $\pi[b]$ is consistent with b. The redistribution process of Section 4.2 is a first step in this direction: The orthogonal projection is the result of a more "cautious" approach (with respect to $BetP$) in which the mass of higher-size events is not divided among singletons, but among subsets.

References

1. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
2. Yaghlane, A.B., Denoeux, T., Mellouli, K.: Coarsening approximations of belief functions. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, pp. 362–373. Springer, Heidelberg (2001)
3. Cobb, B., Shenoy, P.: On the plausibility transformation method for translating belief function models to probability models. International Journal of Approximate Reasoning 41(3), 314–330 (2006)
4. Denoeux, T.: Inner and outer approximation of belief structures using a hierarchical clustering approach. Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9(4), 437–460 (2001)
5. Denoeux, T., Yaghlane, A.B.: Approximating the combination of belief functions using the fast Moebius transform in a coarsened frame. International Journal of Approximate Reasoning 31(1-2), 77–101 (2002)

6. Haenni, R., Lehmann, N.: Resource bounded and anytime approximation of belief function computations. International Journal of Approximate Reasoning 31(1-2), 103–154 (2002)
7. Bauer, M.: Approximation algorithms and decision making in the Dempster-Shafer theory of evidence–an empirical study. International Journal of Approximate Reasoning 17, 217–237 (1997)
8. Tessem, B.: Approximations for efficient computation in the theory of evidence. Artificial Intelligence 61(2), 315–329 (1993)
9. Lowrance, J.D., Garvey, T.D., Strat, T.M.: A framework for evidential-reasoning systems. In: American Association for Artificial Intelligence, (ed.) Proceedings of the National Conference on Artificial Intelligence, pp. 896–903 (1986)
10. Voorbraak, F.: A computationally efficient approximation of Dempster-Shafer theory. International Journal on Man-Machine Studies 30, 525–536 (1989)
11. Smets, P.: Belief functions versus probability functions. In: Bouchon, B., Saitta, L., Yager, R. (eds.) Uncertainty and Intelligent Systems, pp. 17–24. Springer, Heidelberg (1988)
12. Ha, V., Haddawy, P.: Geometric foundations for interval-based probabilities. In: Cohn, A.G., Schubert, L., Shapiro, S.C. (eds.) KR'98: Principles of Knowledge Representation and Reasoning, pp. 582–593. Morgan Kaufmann, San Francisco (1998)
13. Black, P.: An examination of belief functions and other monotone capacities. PhD dissertation, Department of Statistics, CMU (1996) Pgh. PA 15213
14. Cozman, F.G.: Calculation of posterior bounds given convex sets of prior probability measures and likelihood functions. Journal of Computational and Graphical Statistics 8(4), 824–838 (1999)
15. Berger: Robust bayesian analysis: Sensitivity to the prior. Journal of Statistical Planning and Inference 25, 303–328 (1990)
16. Herron, T., Seidenfeld, T., Wasserman, L.: Divisive conditioning: further results on dilation. Philosophy of Science 64, 411–444 (1997)
17. Seidenfeld, T., Wasserman, L.: Dilation for convex sets of probabilities. Annals of Statistics 21, 1139–1154 (1993)
18. Cuzzolin, F.: Geometry of upper probabilities. In: Proceedings of the 3^{rd} Internation Symposium on Imprecise Probabilities and Their Applications (ISIPTA'03) (July 2003)
19. Cuzzolin, F.: A geometric approach to the theory of evidence. IEEE Transactions on Systems, Man and Cybernetics part C (to appear)
20. Dubrovin, B., Novikov, S., Fomenko, A.: Geometria Contemporanea 3. Editori Riuniti (1989)
21. Chateauneuf, A., Jaffray, J.Y.: Some characterizations of lower probabilities and other monotone capacities through the use of Möbius inversion. Mathematical Social Sciences 17, 263–283 (1989)
22. Dubois, D., Prade, H., Smets, P.: A definition of subjective possibility. International Journal of Approximate Reasoning (in press, March 2007)
23. Smets, P.: Decision making in the TBM: the necessity of the pignistic transformation. International Journal of Approximate Reasoning 38(2), 133–147 (2005)
24. Cuzzolin, F.: Two new Bayesian approximations of belief functions based on convex geometry. IEEE Transactions on Systems, Man and Cybernetics part B (2007)
25. Smets, P.: The nature of the unnormalized beliefs encountered in the Transferable Belief Model. In: Proceedings of UAI'92, San Mateo, CA, p. 292. Morgan Kaufmann, San Francisco (1992)

On Latent Belief Structures*

Frédéric Pichon[1,2] and Thierry Denœux[1]

[1] UMR CNRS 6599 Heudiasyc, Université de Technologie de Compiègne,
BP20529, F-60205 Compiègne Cedex, France
{frederic.pichon,thierry.denoeux}@hds.utc.fr
[2] Thales Research and Technology,
RD 128, F-91767 Palaiseau cedex, France

Abstract. Based on the canonical decomposition of belief functions, Smets introduced the concept of a latent belief structure (LBS). This concept is revisited in this article. The study of the combination of LBSs allows us to propose a less committed version of Dempster's rule, resulting in a commutative, associative and idempotent rule of combination for LBSs. This latter property makes it suitable to combine non distinct bodies of evidence. A sound method based on the plausibility transformation is also given to infer decisions from LBSs. In addition, an extension of the new rule is proposed so that it may be used to optimize the combination of imperfect information with respect to the decisions inferred.

1 Introduction

The theory of belief functions [14] is recognized as a rich framework for representing and reasoning with imperfect information. Contrary to probability theory, it allows in particular the representation of different forms of ignorance. However, when decisions have to be made in an uncertain context, rationality principles [13] justify the use of a probability distribution. There exist different methods for the transformation of a belief function to a probability distribution; in particular the pignistic transformation [17] and the plausibility transformation [2]. In this article, two results related to the latter transformation are presented. First, it can be extended to transform a so-called latent belief structure (LBS) [16] into a probability distribution. Second, two ways of modeling negative statements become equivalent with the extension of this transformation.

Equipped with a well-defined means to use LBSs with respect to decision making, this paper deepens their study. The analysis of the combination of LBSs leads to families of conjunctive combination rules. One of these rules is idempotent, a property required for the combination of LBSs obtained from, e.g., belief functions based on non distinct bodies of evidence.

The rest of this paper is organized as follows. The mathematical concept of LBS and Smets's tentative interpretation of a LBS will first be recalled in Section 2. Combination rules for LBSs will then be studied in Section 3. Section 4 will describe decision making from LBSs and Section 5 will conclude the paper.

* The authors would like to thank the anonymous referees for their constructive comments that helped to improve this paper.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 368–380, 2007.

2 Latent Belief Structures

2.1 Background Material on Belief Function Theory

The presentation of belief function theory adopted here is in line with the one of the transferable belief model (TBM) [17]. The beliefs held by a rational agent Ag on a finite frame of discernment $\Omega = \{\omega_1, ..., \omega_K\}$ are represented by a basic belief assignment (BBA) m^{Ω}_{Ag} defined as a mapping from 2^{Ω} to $[0, 1]$ verifying $\sum_{A \subseteq \Omega} m(A) = 1$. For $A \subseteq \Omega$, if $m(A) > 0$ holds, then A is called a focal set (FS) of m. A BBA m is called: *normal* if $\emptyset$ is not a FS; *vacuous* if Ω is the only FS; *dogmatic* if Ω is not a FS; *categorical* if is has only one FS different from Ω; *simple* if it has at most two FSs, Ω included. If m is a simple BBA (SBBA) defined by $m(A) = 1 - w$ and $m(\Omega) = w$ for $A \neq \Omega$, it is noted A^w; if $A = \Omega$ then we write Ω if no confusion can occur. Note that normality is not required by the TBM. Equivalent representations of a BBA m exist. In particular the belief, plausibility, and commonality functions are defined, respectively, by:

$$bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B) \quad , \tag{1}$$

$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B) \quad , \tag{2}$$

and

$$q(A) = \sum_{B \supseteq A} m(B) \quad , \tag{3}$$

for all $A \subseteq \Omega$. Two distinct BBAs m_1 and m_2 can be combined using the TBM conjunctive combination rule, noted $\textcircled{\cap}$, or using Dempter's rule [14], noted $\oplus$. Assuming that $m_{1\textcircled{\cap}2}(\emptyset) \neq 1$, those rules are defined by:

$$m_{1\textcircled{\cap}2}(A) = \sum_{B \cap C = A} m_1(B)\, m_2(C), \forall A \subseteq \Omega \quad , \tag{4}$$

$$m_{1\oplus 2}(A) = \begin{cases} 0 & \text{if } A = \emptyset, \\ m_{1\textcircled{\cap}2}(A) \,/\, \left(1 - m_{1\textcircled{\cap}2}(\emptyset)\right) & \text{otherwise.} \end{cases} \tag{5}$$

Under conflicting information, i.e. $m_{1\textcircled{\cap}2}(\emptyset) > 0$, the legitimacy of the normalization operation involved by Dempster's rule has been questioned. Indeed, the conflict may originate from different situations such as unreliable sources of information or a lack of exhaustiveness of Ω, in which cases other normalization operations may be reasonable [6].

The pignistic and the plausibility transformations allow the transformation of a BBA m to probability distributions noted respectively $BetP_m$ and PlP_m. They are defined as follows:

$$BetP_m(\{\omega_k\}) = \sum_{\{A \subseteq \Omega, \omega_k \in A\}} \frac{m(A)}{(1 - m(\emptyset))\,|A|} \quad , \tag{6}$$

$$PlP_m(\{\omega_k\}) = \kappa^{-1} pl(\{\omega_k\}) \, , \tag{7}$$

with $\kappa = \sum_{j=1}^{K} pl(\{\omega_j\})$.

2.2 Canonical Decomposition of a Belief Function

The canonical decomposition of a belief function, introduced in [16], is based on a generalization of the canonical representation of a *separable* BBA m defined by Shafer [14]. A BBA is called separable if it can be written as the $\textcircled{\cap}$ combination of SBBAs. For a separable BBA m, one has thus:

$$m = \textcircled{\cap}_{A \subset \Omega} A^{w(A)}, \tag{8}$$

with $w(A) \in [0,1]$ for all $A \subset \Omega$. Through the definition of a *generalized* SBBA, Smets [16] proposed a means to canonically decompose any *non dogmatic* BBA (NDBBA). A generalized SBBA is defined as a function μ from 2^Ω to $\mathbb{R}$ by:

$$\mu(A) = 1 - w,\ \mu(\Omega) = w,\ \mu(B) = 0 \ \ \forall B \in 2^\Omega \backslash \{A, \Omega\}, \tag{9}$$

for $A \neq \Omega$ and $w \in [0, +\infty)$. Extending the SBBA notation, any generalized SBBA can be written A^w; when $w \leq 1$, μ is thus a SBBA. When $w > 1$, μ is called *inverse* SBBA. Smets showed that any NDBBA can be uniquely represented as the $\textcircled{\cap}$ combination of non categorical generalized SBBAs; the expression for this decomposition is then the same as (8) with $w \in (0, +\infty)$ this time. The weights $w(A)$ for each $A \in 2^\Omega \backslash \{\Omega\}$ are obtained as follows:

$$w(A) = \prod_{B \supseteq A} q(B)^{(-1)^{|B|-|A|+1}}. \tag{10}$$

The weight function, $w : 2^\Omega \backslash \{\Omega\} \rightarrow (0, +\infty)$, is thus yet another equivalent representation of a NDBBA m.

If A^{w_1} and A^{w_2} are two SBBAs, their combination by $\textcircled{\cap}$ is the SBBA $A^{w_1 w_2}$. From the commutativity and associativity of the $\textcircled{\cap}$ rule, the combination of two NDBBAs m_1 and m_2 with respective weight functions w_1 and w_2 is written:

$$m_1 \textcircled{\cap} 2 = \textcircled{\cap}_{A \subset \Omega} A^{w_1(A) \cdot w_2(A)}. \tag{11}$$

Details on normalized versions of those results can be found in a recent exposition of the canonical decomposition [3]. Other combinations of belief functions have been proposed. In particular the cautious rule [3], noted $\textcircled{\wedge}$, possesses the idempotence property. It is defined as follows ($\wedge$ is the minimum operator):

$$m_1 \textcircled{\wedge} 2 = \textcircled{\cap}_{A \subset \Omega} A^{w_1(A) \wedge w_2(A)}. \tag{12}$$

2.3 Decombination Rule

In the area of belief revision [8], the addition of beliefs without retracting others is known as *expansion*; the inverse operation, *contraction*, allows the removal of beliefs. In belief function theory, those operations are performed respectively by the $\textcircled{\cap}$ and $\textcircled{\not\cap}$ rules. Different authors [16,15,10] have studied the $\textcircled{\not\cap}$ rule which is either called the *decombination* [16] or *removal* [15] rule. Let q_1 and q_2 be the commonality functions of two NDBBAs, the decombination is defined by:

$$q_1 \textcircled{\not\cap} 2\,(A) = q_1(A) / q_2(A), \quad \forall A \subseteq \Omega. \tag{13}$$

The resulting function may not be a belief function. In this case it is called a *pseudo* belief function [16] or *signed* belief function [10].

The interest of this operator is motivated by the following example. Suppose you are in a state of ignorance about the actual state ω_0 of the world Ω. Suppose that you then have some good reasons to believe in A, for $A \subset \Omega$; you perform an *expansion* of your beliefs (here your ignorance) by the SBBA A^x (if the *good reasons* amounts to $(1-x)$, for x small), which is equivalent to the combination $\Omega \textcircled{\cap} A^x$. Later, another information arrives telling you that the first information was not valid. This can be handled in at least three ways, all of them bringing you to the state of indecision that this example intuitively leads to. Note that this situation is illustrated by *the Pravda bias* example [16].

First of all, the second information can be understood as: the negation of the first information holds true. This results in an expansion of belief in favor of $\bar{A}$. It leads to the state of belief $A^x \textcircled{\cap} \bar{A}^x$, yielding $bel(A) = bel\left(\bar{A}\right)$ which is indeed a state of indecision. This solution produces however a share of conflict ($m(\emptyset) > 0$) depending on the value of x. This conflict cannot be escaped whatever further information you receive [10], unless an arbitrary normalization operation is used.

Another way of interpreting the second information is that all conclusions that may be drawn from the first information must be cancelled, i.e. you should come back to the state of belief in which you were before receiving the first information [10]. This means here that you should come back to the state of ignorance, i.e. $bel(\Omega) = 1$, which is a state of indecision. This interpretation of the example may be treated in two ways. It may be argued that both ways involve a contraction in their development as showed by the following reasoning.

The first way of treating this second interpretation of the example consists in contracting the belief you had in favor of A, which amounts to do $A^x \textcircled{\oslash} A^x = \Omega$. The ignorance state is thus recovered. The second method uses the discounting operation [14]. This operation is based on the use of a second frame $\mathcal{R} = \{R, NR\}$ which represents a meta-knowledge $m^{\mathcal{R}}$ on the reliability of the information that is given to you. If the first source of information is reliable then your belief on Ω is the one given by this source; this is noted $m^{\Omega}_{Ag}[\{R\}] = m^{\Omega}_{S}$ where S denotes the first source of information. If the source of information is not reliable then your belief is vacuous: $m^{\Omega}_{Ag}[\{NR\}](\Omega) = 1$. Let us suppose that before you receive the second information, you are a priori almost certain that S is reliable: $m^{\mathcal{R}}_{Ag}(\{R\}) = 1 - \epsilon$, and $m^{\mathcal{R}}_{Ag}(\mathcal{R}) = \epsilon$, with ϵ a small positive real number. Your belief on Ω is computed by $\textcircled{\cap}$ combining $m^{\mathcal{R}}_{Ag}$ with $m^{\Omega}_{Ag}[\{R\}]$ and then marginalizing this belief on Ω. To be in a state of indecision after receiving the second information, $m^{\mathcal{R}}_{Ag}(\{R\}) = 0$ must hold. It is possible by a contraction of your initial $m^{\mathcal{R}}_{Ag}$. It is also possible through the $\textcircled{\cap}$ combination of your initial $m^{\mathcal{R}}_{Ag}$ with a BBA $m^{\mathcal{R}}(\{NR\}) = 1$; this solution implies however the use of a categorical belief (see [3] for a discussion on dogmatic beliefs).

More generally, the $\textcircled{\oslash}$ rule allows a form of non monotonic reasoning in the belief function theory. Indeed if for $A \subseteq \Omega$ you have a belief $bel(A) > 0$ then it will be impossible without this operator to obtain later $bel\left(\bar{A}\right) = 1$ by an expansion with other beliefs, unless normalization is used.

Smets [16] goes further than the interpretation of removal of beliefs that is given to the $\textcircled{\scriptsize /}$ operator. He introduces states of debt of belief (also called *diffident* beliefs). Indeed, Smets reformulates the example so that the second source of information gives good reasons *not to believe* in A. He argues that if the weights of the good reasons to believe A and not to believe A counterbalance each other, then you should be in a state of ignorance. Those states of debt of belief are used by Smets to introduce the LBSs recalled in the next subsection.

Note that the existence of positive and negative information is generally coined under the term *bipolarity*. Other authors have tried to model such dual information in the belief function theory; in particular we can cite the work of Dubois et al. [7], and of Labreuche et al. [11]. The question of the relevance of debt of belief remains open. Nonetheless, the next subsection shows that decombination is useful at least mathematically and thus deserves attention.

2.4 Confidence and Diffidence

Let A^{w_1} and A^{w_2} be two non categorical SBBAs, hence A^{1/w_2} is an inverse SBBA. The decombination of A^{w_1} by A^{w_2}, i.e. $A^{w_1}\textcircled{\scriptsize /}A^{w_2}$, is equal to the $\textcircled{\cap}$ combination of A^{w_1} with A^{1/w_2} [16]. Let w be the weight function of a NDBBA m. Partition 2^Ω into two (disjoint) subsets: $C = \{A : A \subset \Omega, w(A) \in (0,1]\}$, and $D = \{A : A \subset \Omega, w(A) \in (1,\infty)\}$. A NDBBA m can then be written:

$$m = \left(\textcircled{\cap}_{A\in C} A^{w(A)}\right) \textcircled{\scriptsize /} \left(\textcircled{\cap}_{A\in D} A^{\frac{1}{w(A)}}\right) \tag{14}$$

Any NDBBA is thus the result of combinations and decombinations of non categorical SBBAs or, equivalently, any NDBBA is equal to the decombination of a separable NDBBA by a separable NDBBA. Smets called the separable NDBBA, noted m^c and obtained from the set C, the *confidence* component and the separable NDBBA, noted m^d and obtained from the set D, the *diffidence* component. We can thus write: $m = m^c \textcircled{\scriptsize /} m^d$. The weight functions of m^c and m^d, defined from $2^\Omega \setminus \{\Omega\}$ to $(0,1]$ and called the confidence and diffidence weight functions, are noted w^c and w^d. They can easily be found from the original weight function w of a NDBBA m as follows: $w^c(A) = 1 \wedge w(A)$, and $w^d(A) = 1 \wedge \frac{1}{w(A)}$, for all $A \subset \Omega$.

From the canonical decomposition of a belief function, Smets defined a LBS as a pair of BBAs $\left(m^c, m^d\right)$ allowing the representation of belief states in which positive and negative items of evidence (reasons *to believe and not to believe* [16]) occur. Definition 1 is more specific, in that it imposes that this pair be made of separable NDBBAs. Definition 2 defines a concept also introduced in [16].

Definition 1 (Latent Belief Structure). *A latent belief structure is defined as a pair of separable NDBBAs m^c and m^d called respectively the confidence and diffidence components. A LBS is noted using a upper-case L.*

Definition 2 (Apparent Belief Structure). *The apparent belief structure associated with a LBS $L = (m^c, m^d)$ is the signed belief function obtained from the decombination $m^c \textcircled{\scriptsize /} m^d$ of the confidence and diffidence components of L.*

The motivation for Definition 1 is due to the following observation: if m^c and m^d are two NDBBAs, then we can always find two separable NDBBAs m'^c and m'^d such that $m^c \textcircled{@} m^d = m'^c \textcircled{@} m'^d$, hence a LBS can be merely defined as a pair of separable NDBBAs. A LBS is thus a generalization of a NDBBA.

The properties linking these definitions are the following. By definition, the apparent belief structure associated to a LBS may or may not be a belief function. Further, an infinity of LBSs correspond to the same apparent belief structure. Besides, among the infinity of LBSs corresponding to the same apparent belief structure, one LBS has a particular structure: it is then called a *canonical* LBS (see Definition 3 below). In particular, an infinity of LBSs can yield the same NDBBA, for instance the LBSs $(A^{0.2}, A^{0.3})$ and $(A^{0.6}, A^{0.9})$ correspond to the same NDBBA $A^{2/3}$ and the canonical LBS of this NDBBA is $\left(A^{2/3}, \Omega\right)$. Remark that the canonical decomposition of a NDBBA m yields the canonical LBS of m. Example 1 shows how a CLBS can be generated from expert opinions. Note that L_Ω will be used to denote the LBS obtained from the vacuous BBA.

Definition 3 (Canonical Latent Belief Structure). *A CLBS is a LBS verifying:* $\forall A \subset \Omega$, $w^c(A) \vee w^d(A) = 1$ *where* $\vee$ *denotes the maximum operator.*

Example 1. Suppose $\Omega = \{a, b, c\}$ and the sets $A = \{a, b\}, B = \{b, c\}, C = \{a, c\}$. Now, a first expert gives the opinion $A \sim B$ which, according to the elicitation technique proposed in [1], means that he believes equivalently in A and B, i.e. $bel_1(A) = bel_1(B)$. A second expert gives the opinion: $C \sim B$, i.e $bel_2(C) = bel_2(B)$. Given those constraints on bel_1 and bel_2, the BBAs m_1 and m_2 of Table 1 on page 7 may be produced using the method proposed in [1] for a certain set of parameters required by the method.

3 Combination of LBSs

This section studies mathematical operations on LBSs. Let us first express two known operations of belief function theory using LBSs.

Let $\left(m_1^c, m_1^d\right)$ and $\left(m_2^c, m_2^d\right)$ be the CLBSs associated with two NDBBAs m_1 and m_2. Then $\left(m_1^c \textcircled{\cap} m_2^c, m_1^d \textcircled{\cap} m_2^d\right)$ is a LBS associated with $m_1 \textcircled{\cap} m_2$. This lead Smets to define the conjunctive combination of two LBSs as follows.

Definition 4. *The conjunctive combination of two LBSs* L_1 *and* L_2 *is a LBS noted* $L_{1\textcircled{\cap}2}$. *It is defined by the weight functions (15) and (16):*

$$w^c_{1\textcircled{\cap}2}(A) = w^c_1(A) \cdot w^c_2(A), \tag{15}$$

$$w^d_{1\textcircled{\cap}2}(A) = w^d_1(A) \cdot w^d_2(A). \tag{16}$$

The vacuous LBS L_Ω is a neutral element for $\textcircled{\cap}$, i.e. $L \textcircled{\cap} L_\Omega = L$ for all LBSs L. The cautious rule of combination [3] can also be expressed in terms of LBSs.

Definition 5. *([4, Proposition 6]) The cautious combination of two LBSs* L_1 *and* L_2 *is a LBS noted* $L_{1\textcircled{\wedge}2}$. *It is defined by the following weight functions:*

$$w^c_{1\textcircled{\wedge}2}(A) = w^c_1(A) \wedge w^c_2(A), \tag{17}$$

$$w^d_{1\textcircled{\wedge}2}(A) = w^d_1(A) \vee w^d_2(A). \tag{18}$$

Table 1. Two NDBBAs and their CLBSs

A	$m_1(A)$	$w_1^c(A)$	$w_1^d(A)$	$m_1^c(A)$	$m_1^d(A)$	$m_2(A)$	$w_2^c(A)$	$w_2^d(A)$	$m_2^c(A)$	$m_2^d(A)$
$\emptyset$	0	1	1	0	0	0	1	1	0	0
$\{a\}$	0	1	1	0	0	0	1	1	0	0
$\{b\}$	0	1	5/9	4/9	4/9	0	1	1	0	0
$\{a,b\}$	0.4	1/3	1	2/9	0	0	1	1	0	0
$\{c\}$	0	1	1	0	0	0	1	5/9	4/9	4/9
$\{a,c\}$	0	1	1	0	0	0.4	1/3	1	2/9	0
$\{b,c\}$	0.4	1/3	1	2/9	0	0.4	1/3	1	2/9	0
Ω	0.2			1/9	5/9	0.2			1/9	5/9

It is clear that those two rules belong to different families of combination rules: the $\textcircled{\cap}$ rule is purely conjunctive whereas the $\textcircled{\wedge}$ rule is both conjunctive and disjunctive [4], hence they treat the diffidence component in different ways. The remainder of this section is devoted to the proposal of other purely conjunctive rules. One of those rules is particularly interesting since it is idempotent; the motivation for its definition relies on the *least commitment principle* of the TBM.

3.1 Least Commitment Principle (LCP)

The LCP is similar to the principle of maximum entropy in Bayesian Probability Theory. It postulates that given a set $\mathcal{M}$ of BBAs compatible with a set of constraints, the most appropriate BBA is the least informative. This principle becomes operational through the definition of partial orders allowing the informational comparison of BBAs. Such orders, generalizing set inclusion, are [5]:

- pl-order: for all $A \subseteq \Omega$, iff $pl_1(A) \leq pl_2(A)$ then $m_1 \sqsubseteq_{pl} m_2$;
- q-order: for all $A \subseteq \Omega$, iff $q_1(A) \leq q_2(A)$ then $m_1 \sqsubseteq_q m_2$;
- s-order: $m_1 \sqsubseteq_s m_2$, i.e. m_1 is a *specialization* of m_2, iff there exists a square matrix S with general term $S(A,B)$, $A, B \subseteq \Omega$ such that $\sum_{B \subseteq \Omega} S(A,B) = 1$, $\forall A \subseteq \Omega$, and $S(A,B) > 0 \Rightarrow A \subseteq B, \forall A, B \subseteq \Omega$, and $m_1(A) = \sum_{B \subseteq \Omega} S(A,B)\, m_2(A)$, $\forall A \subseteq \Omega$.

A BBA m_1 is said to be x-*more committed* than m_2, with $x \in \{pl, q, s\}$, if $m_1 \sqsubseteq_x m_2$. A particular case of specialization is the dempsterian specialization [9], noted $\sqsubseteq_d$: $m_1 \sqsubseteq_d m_2$, iff there exists a BBA m such that $m_1 = m \textcircled{\cap} m_2$. This condition is stronger than specialization, i.e. $m_1 \sqsubseteq_d m_2 \Rightarrow m_1 \sqsubseteq_s m_2$.

It is reasonable to say that a SBBA A^{w_1} is more committed than a SBBA A^{w_2}, if $w_1 \leq w_2$. Hence a BBA m_1 obtained from the combination by $\textcircled{\cap}$ of SBBAs, i.e. a separable BBA, will be more committed than another separable BBA m_2 if $w_1(A) \leq w_2(A)$ for all $A \in 2^\Omega \backslash \{\Omega\}$; this is equivalent to the existence of a separable BBA m such that $m_1 = m \textcircled{\cap} m_2$. This new partial order, defined for separable BBAs and noted $m_1 \sqsubseteq_w m_2$ with m_1 and m_2 two separable BBAs, is consequently stricly stronger than d-ordering as a non-separable BBA m such that $m_1 = m \textcircled{\cap} m_2$, i.e. $m_1 \sqsubseteq_d m_2$, can easily be found. Let us also

remark that, using a special representation of categorical BBAs, Denoeux [4, Proposition 3] has shown that $\sqsubseteq_w$ may be seen as generalizing set inclusion, much as the x-orderings, with $x \in \{pl, q, s\}$, do.

All those partial orders can be extended to LBSs. In particular, the LBS $L_{1\textcircled{\cap}2} = (m^c_{1\textcircled{\cap}2}, m^d_{1\textcircled{\cap}2})$ obtained from the combination by $\textcircled{\cap}$ of two LBSs $L_1 = (m^c_1, m^d_1)$ and $L_2 = (m^c_2, m^d_2)$ has the following properties: for $i \in \{1, 2\}$, $m^c_{1\textcircled{\cap}2} \sqsubseteq_w m^c_i$, and $m^d_{1\textcircled{\cap}2} \sqsubseteq_w m^d_i$, i.e. the $\textcircled{\cap}$ rule produces a LBS $L_{1\textcircled{\cap}2}$ which is both w-more committed in confidence and in diffidence than the LBSs L_1 and L_2. To simplify the presentation, a LBS L which is both w-more committed in confidence and in diffidence than a LBS L' will be noted $L \sqsubseteq_l L'$ (l stands for *latent*). Informally, the l-order seems natural as it exhibits properties similar to those of classical orderings. To see that, simply replace l by x with $x \in \{d, s, q, pl\}$ and L by m in the following expressions: $\forall L, L \sqsubseteq_l L_\Omega$ and $L_{1\textcircled{\cap}2} \sqsubseteq_l L_1$, $L_{1\textcircled{\cap}2} \sqsubseteq_l L_2$.

3.2 Combination of Non-distinct LBSs

As remarked in [5], it is possible to think of $\sqsubseteq_x$ as generalizing set inclusion. This reasoning can be used to see conjunctive combination rules as generalizing set intersection. Denœux [3] considers thus the following situation. Suppose we get two reliable sources of information. One states that ω is in $A \subseteq \Omega$, whereas the other states that it is in $B \subseteq \Omega$. It is then certain that ω is in C such that $C \subseteq A$ and $C \subseteq B$. The largest subset C satisfying those constraints is $A \cap B$.

Suppose now that the sources provide the NDBBAs m_1 and m_2 and let L_1 and L_2 be the equivalent CLBSs of those NDBBAs. Upon receiving those two pieces of information, the agent's state of belief should be represented by a LBS L_{12}, i.e. $\left(m^c_{12}, m^d_{12}\right)$, more informative than L_1 and L_2. Let $\mathcal{S}_x(L)$ be the set of LBSs L' such that $L' \sqsubseteq_x L$. Hence $L_{12} \in \mathcal{S}_x(L_1)$ and $L_{12} \in \mathcal{S}_x(L_2)$, or equivalently $L_{12} \in \mathcal{S}_x(L_1) \cap \mathcal{S}_x(L_2)$. According to the LCP, the x-least committed LBS should be chosen in $\mathcal{S}_x(L_1) \cap \mathcal{S}_x(L_2)$. This defines a conjunctive combination rule if the x-least committed LBS exists and is unique. Proposition 1 shows that the l-order may be an interesting solution for this problem.

Proposition 1. *Let L_1 and L_2 be two LBSs. The l-least committed element in $\mathcal{S}_l(L_1) \cap \mathcal{S}_l(L_2)$ exists and is unique (the proof is trivial by Proposition 1 of [3]). It is defined by the following confidence and diffidence weight functions:*

$$w^c_{1\textcircled{\wedge}2}(A) = w^c_1(A) \wedge w^c_2(A), \quad A \in 2^\Omega \backslash \{\Omega\}, \tag{19}$$

$$w^d_{1\textcircled{\wedge}2}(A) = w^d_1(A) \wedge w^d_2(A), \quad A \in 2^\Omega \backslash \{\Omega\}. \tag{20}$$

Definition 6 (Weak Rule). *Let L_1 and L_2 be two LBSs. Their combination with the weak rule is defined as the LBS whose weight functions are given by (19) and (20). It is noted: $L_{1\textcircled{\wedge}2}$.*

This rule is commutative, associative and idempotent. In addition, $\textcircled{\cap}$ is distributive with respect to $\textcircled{\wedge}$. Those properties originate from the properties of the $\textcircled{\wedge}$ rule [3] since there is a connection between the partial orders on which those two

rules are built. We can thus see that the combination by the $\circledwedge$ rule consists in combining the confidence and diffidence components by the $\circledwedge$ rule.

The $\circledwedge$ rule exhibits other properties: L_Ω is a neutral element and if $L_1 \sqsubseteq_l L_2$, the result of the least committed combination of those LBSs is $L_1 \circledwedge L_2 = L_1$. Further, using the l-order in the derivation of the rule allows the construction of a 'weaker', or l-less committed, version of Dempster's rule, i.e. $L_1 \circledcap L_2 \sqsubseteq_l L_1 \circledwedge L_2$.

Note that the apparent form of a LBS $L_{1\circledwedge 2}$, produced by the $\circledwedge$ combination of two CLBSs L_1 and L_2 obtained from two NDBBAs m_1 and m_2, may not be a BBA. However, if m_1 and m_2 are separable BBAs then the apparent form of the LBS $L_{1\circledwedge 2}$ is a BBA since a separable BBA yields a LBS whose diffidence component is vacuous and the $\circledwedge$ combination consists in combining the confidence component of L_1 and L_2 by the $\circledwedge$ rule. It can also be shown that the combination by the $\circledwedge$ rule of two consonant BBAs does not always yield a consonant BBA. A consonant BBA is separable [4, Proposition 2], hence the $\circledwedge$ rule applied to two CLBSs obtained from two consonant BBAs will yield a LBS whose apparent form is a separable BBA which is not necessarily consonant.

Example 2. The left-hand part of Table 2 shows the weight functions resulting from the weak ($\circledwedge$), the conjunctive ($\circledcap$), and the cautious ($\circledwedge$) combinations of the expert opinions of Example 1. Note that $w^d_{1\circledwedge 2}(A) = 1$, for all $A \subset \Omega$.

Table 2. Weight functions obtained from different combinations (left). Plausibility transformations of the LBSs obtained with those combinations (right, see Section 4).

A	$w^c_{1\circledwedge 2}(A)$	$w^d_{1\circledwedge 2}(A)$	$w^c_{1\circledcap 2}(A)$	$w^d_{1\circledcap 2}(A)$	$w^c_{1\circledwedge 2}(A)$	$PlP_{1\circledwedge 2}$	$PlP_{1\circledcap 2}$	$PlP_{1\circledwedge 2}$
$\emptyset$	1	1	1	1	1			
$\{a\}$	1	1	1	1	1	9/19	0.23	1/3
$\{b\}$	1	5/9	1	5/9	1	5/19	0.385	1/3
$\{a,b\}$	1/3	1	1/3	1	1/3			
$\{c\}$	1	5/9	1	5/9	1	5/19	0.385	1/3
$\{a,c\}$	1/3	1	1/3	1	1/3			
$\{b,c\}$	1/3	1	1/9	1	1/3			

Interestingly, the idea of distinctness conveyed by the derivation of the $\circledwedge$ rule relates, in part, to the foci of the SBBAs underlying a complex belief state. Indeed let us assume that two bodies of evidence, yielding the LBSs L_1 and L_2, are non distinct, then $L_{1\circledwedge 2} \neq L_{1\circledcap 2}$ iff $C_1 \cap C_2 \neq \emptyset$ or $D_1 \cap D_2 \neq \emptyset$, with $C_i = \{A : A \subset \Omega, w^c_i(A) < 1\}$ and $D_i = \{A : A \subset \Omega, w^d_i(A) < 1\}$. The effect of this view of distinctness is illustrated in Example 3 of Section 4, where double counting the SBBA implicitly shared by two agents is avoided.

3.3 Generalizing the Weak Rule

In the same vein as Denœux [3], it is possible to derive infinite families of conjunctive combination rules for LBSs. The $\circledcap$ and $\circledwedge$ rules are then merely instances of these families. This extension is based on the observation that the $\circledcap$ rule uses the

product, whereas the $\textcircled{\wedge}$ rule uses the minimum of weights belonging to the unit interval. Now, these two operations on this interval are binary operators known as triangular norms (t-norms). Replacing them by any positive t-norm $\top$ yields $\textcircled{\top}$ operators, which possess the following properties: commutativity, associativity, neutral element L_Ω and monotonicity with respect to $\sqsubseteq_l$, i.e. $\forall L_1, L_2$ and L_3, $L_1 \sqsubseteq_l L_2 \Rightarrow L_1 \textcircled{\top} L_3 \sqsubseteq_l L_2 \textcircled{\top} L_3$. Only the $\textcircled{\wedge}$ rule is idempotent. Operators exhibiting a behavior between $\textcircled{\cap}$ and $\textcircled{\wedge}$ can be obtained using parameterized families of t-norms such as the Dubois and Prade family defined by:

$$x \top_\gamma^{DP} y = (xy) \,/\, (\max(x, y, \gamma)) \text{ for } x, y, \text{ and } \gamma \in [0, 1]\,. \tag{21}$$

Note that the $\textcircled{\cap}$ and $\textcircled{\wedge}$ rules are recovered for $\gamma = 1$ and $\gamma = 0$ respectively. The parameterization is what make those rules attractive: they allow the fine-tuning of the behavior of a system. Indeed, the γ parameter may be related to some subjective judgment on the distinctness of the items of evidence. It can also be learnt from data as done in [12] through the use of the plausibility transformation extended to LBSs (see Section 4): the conjunctive combination of two LBSs is then optimized with respect to the decisions inferred.

4 Decision Making with LBSs

This section provides a means to transform a LBS into a probability distribution. The plausibility transformation is of particular interest here due to one of its properties: it is invariant with respect to the combination by $\textcircled{\cap}$ [18], which is not the case of the pignistic transformation. Proposition 2 reformulates this property for the $\textcircled{\eth}$ rule using the decombination operator in probability theory, noted $\oslash$ and defined in [15] as follows. Let P_1 and P_2 be two probability distributions:

$$P_1 \oslash P_2\,(\{\omega_k\}) = \kappa^{-1} P_1\,(\{\omega_k\})\,/P_2\,(\{\omega_k\}) \quad , \forall \omega_k \in \Omega \tag{22}$$

with $\kappa = \sum_{j=1}^{K} P_1\,(\{\omega_j\})\,/P_2\,(\{\omega_j\})$.

Proposition 2 (PlP is invariant with respect to $\textcircled{\eth}$). *Let m_1 and m_2 be two NDBBAs:*

$$PlP_{m_1 \textcircled{\eth} m_2} = PlP_{m_1} \oslash PlP_{m_2}\ . \tag{23}$$

Proof. For all $\omega_k \in \Omega$, let us denote $\alpha_k = pl_1\,(\{\omega_k\}) = q_1\,(\{\omega_k\})$, $\beta_k = pl_2\,(\{\omega_k\})$. From Equation (13) we have:

$$PlP_{m_1 \textcircled{\eth} m_2}\,(\{\omega_k\}) = (\alpha_k/\beta_k)\,/(\sum_{i=1}^{K} (\alpha_i/\beta_i))\ . \tag{24}$$

Besides,

$$PlP_{m_1} \oslash PlP_{m_2}\,(\{\omega_k\}) = \left(\left(\frac{\alpha_k}{\sum_{i=1}^{K}\alpha_i}\right) / \left(\frac{\beta_k}{\sum_{i=1}^{K}\beta_i}\right)\right) / \left(\sum_{j=1}^{K} \frac{\left(\frac{\alpha_j}{\sum_{i=1}^{K}\alpha_i}\right)}{\left(\frac{\beta_j}{\sum_{i=1}^{K}\beta_i}\right)}\right) \tag{25}$$

(24) and (25) are equal. □

Using Proposition 2, a LBS $L = \left(m^c, m^d\right)$ can be transformed into a probability distribution as follows:

$$PlP_L = PlP_{m^c} \oslash PlP_{m^d} \ . \tag{26}$$

Example 3. The right side of Table 2 shows three qualitatively different probability distributions computed using (26). They are obtained from the expert opinions of Example 1 combined with different combination rules. It is interesting to note that the application of the $\textcircled{\cap}$ and $\textcircled{\curlywedge}$ rule yields opposite decisions. This is easily explained through the observation of the SBBAs underlying the two opinions. We see in particular that both opinions share a SBBA focused on the set $\{b, c\}$: if we think of the opinions as based on distinct bodies of evidence, then the reasons for which expert 1 believes in $\{b, c\}$ are different from the reasons of expert 2, hence the combined belief in favor of the set $\{b, c\}$ should be stronger than the individual beliefs. On the other hand, if the experts base their beliefs in $\{b, c\}$ on the same items of evidence then the combined belief in favor of the set $\{b, c\}$ should not be stronger than the individual beliefs. Consequently with the $\textcircled{\cap}$ rule we have $w(\{b, c\}) < w(\{a, c\}) = w(\{a, b\})$ which makes the two singletons b and c more probable, actually much more probable than a. This difference is then only partially moderated by the diffidence in b and c so that eventually b and c remain more probable than a. However, with the $\textcircled{\curlywedge}$ and the $\textcircled{\wedge}$ rules, we have $w(\{b, c\}) = w(\{a, c\}) = w(\{a, b\})$, which yields equiprobability for the three singletons. Besides, the $\textcircled{\curlywedge}$ rule keeps the information relating to the diffidence in b and c, hence a is more probable than b and c with this rule.

Proposition 3 shows that two ways of modeling negative statements become equivalent when PlP is used. Indeed, according to Smets's vocabulary [16], for $A \subset \Omega$, *having good reasons to believe in not A* is equivalent to *having good reasons not to believe (or having a debt of belief) in A*. It can also be formulated using the terminology used in belief revision: the expansion by $\bar{A}^\alpha$ is equivalent to the contraction by A^α, for $\alpha \in (0, 1]$. Let $\overset{PlP}{\sim}$ denote the equivalence relation between LBSs defined by $L_1 \overset{PlP}{\sim} L_2$ iff $PlP_{L_1}(\{\omega_k\}) = PlP_{L_2}(\{\omega_k\})$, $\forall \omega_k \in \Omega$.

Proposition 3. $\bar{A}^\alpha \overset{PlP}{\sim} A^{\frac{1}{\alpha}}$, *for* $\alpha \in (0, 1]$.

Proof. $\forall \omega_k \in A, A \subset \Omega$,

$$PlP_{\bar{A}^\alpha}(\{\omega_k\}) = \frac{\alpha}{\left|\bar{A}\right| + |A|\,\alpha} \ , \tag{27}$$

$$PlP_{A^{1/\alpha}}(\{\omega_k\}) = \frac{1}{|A| + \left|\bar{A}\right|\frac{1}{\alpha}} \ . \tag{28}$$

(27) and (28) are equal. □

Propositions 2 and 3 define equivalence classes with respect to the plausibility transformation in which there is at least one separable BBA; for instance we have: $\left(\bar{A}^{0.6}, A^{0.5}\right) \overset{PlP}{\sim} \left(\bar{A}^{0.3}, \Omega\right)$. Note also that the combination by $\textcircled{\cap}$ of any two LBSs

belonging to two different equivalence classes always falls in the same equivalence class, for instance if $L_1 \overset{PlP}{\sim} L_2$ and $L_3 \overset{PlP}{\sim} L_4$, then e.g. $L_1 \textcircled{\cap} L_3 \overset{PlP}{\sim} L_2 \textcircled{\cap} L_4$. It can easily be shown that this is not true for the $\textcircled{\wedge}$ and $\textcircled{\curlywedge}$ rules.

Eventually, from Proposition 3, remark that $A^\alpha \textcircled{\cap} \bar{A}^\alpha \overset{PlP}{\sim} A^\alpha \textcircled{\not\cap} A^\alpha, \forall A \subseteq \Omega$ with $\alpha \in (0, 1]$. Now, let $\overset{BetP}{\sim}$ denote the equivalence relation between BBAs defined by $m_1 \overset{BetP}{\sim} m_2$ iff $BetP_{m_1}(\{\omega_k\}) = BetP_{m_2}(\{\omega_k\})$, for all $\omega_k \in \Omega$. The two ways of modeling negative statements will yield the same probability distribution, i.e. $A^\alpha \textcircled{\cap} \bar{A}^\alpha \overset{BetP}{\sim} A^\alpha \textcircled{\not\cap} A^\alpha$, with the pignistic transformation iff $|A| = |\bar{A}|$; a stricter condition than the one of the plausibility transformation.

5 Conclusion

In this article, latent belief structures have been revisited. The mathematical simplicity of this generalization of non dogmatic belief functions has allowed the analysis of the unnormalized version of Dempster's rule, which resulted in the introduction of infinite families of conjunctive combination rules. Two potential uses of these rules have been proposed. First they may permit to relax the hypothesis of distinctness inherent to the use of Dempster's rule. Second, they may be used to optimize the combination of imperfect information with respect to the decisions inferred. An extension of the plausibility transformation has been also provided to transform a LBS into a probability distribution and two ways of modeling negative statements were proved equivalent under this extension. The interest of this formalism in concrete applications is currently being investigated.

References

1. Ben Yaghlane, A., Denoeux, T., Mellouli, K.: Elicitation of expert opinions for constructing belief functions. In: Proc. of IPMU'2006, vol. 1, pp. 403–411(2006)
2. Cobb, B.R., Shenoy, P.P.: On the plausibility transformation method for translating belief function models to probability models. International Journal of Approximate Reasoning 41, 314–330 (2006)
3. Denœux, T.: The cautious rule of combination for belief functions and some extensions. In: Proceedings of FUSION'2006, Italy, pp. 1–8 (2006)
4. Denœux, T.: Conjunctive and Disjunctive Combination of Belief Functions Induced by Non Distinct Bodies of Evidence. Artificial Intelligence (2007)
5. Dubois, D., Prade, H.: A set-theoretic view of belief functions: logical operations and approximations by fuzzy sets. Int. J. of General Systems 12, 193–226 (1986)
6. Dubois, D., Prade, H.: Representation and combination of uncertainty with belief functions and possibility measures. Computational Intelligence 4, 244–264 (1998)
7. Dubois, D., Prade, H., Smets, Ph.: "Not Impossible" vs. "Guaranteed Possible" in Fusion and Revision. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, pp. 522–531. Springer, Heidelberg (2001)
8. Gärdenfors, P.: Knowledge in Flux: Modeling the Dynamics of Epistemic States. MIT Press, Cambridge (1988)
9. Klawonn, F., Smets, Ph.: The dynamic of belief in the transferable belief model and specialization-generalization matrices. In: Proc. of UAI' 1992, pp. 130–137 (1992)

10. Kramosil, I.: Probabilistic Analysis of Belief Functions. Kluwer Academic Publishers, Dordrecht (2001)
11. Labreuche, Ch., Grabisch, M.: Modeling positive and negative pieces of evidence in uncertainty. In: Nielsen, T.D., Zhang, N.L. (eds.) ECSQARU 2003. LNCS (LNAI), vol. 2711, pp. 279–290. Springer, Heidelberg (2003)
12. Mercier, D., Quost, B., Denoeux, T.: Refined modeling of sensor reliability in the belief function framework using contextual discounting. Information Fusion (2006)
13. Savage, L.J.: Foundation of statistics. Wiley, New York (1954)
14. Shafer, G.: A mathematical theory of evidence. Princeton University Press, Princeton (1976)
15. Shenoy, P.P.: Conditional independence in valuation-based systems. International Journal of Approximate Reasoning 10, 203–234 (1994)
16. Smets, Ph.: The canonical decomposition of a weighted belief. In: Int. Joint Conf. on Artificial Intelligence, San Mateo, pp. 1896–1901. Morgan Kaufman, San Francisco (1995)
17. Smets, Ph.: The Transferable Belief Model for quantified belief representation. In: Gabbay, D., Smets, Ph. (eds.) Handbook of Defeasible reasoning and uncertainty management systems, vol. 1, pp. 264–301. Kluwer Academic Pub., Dordrecht (1998)
18. Voorbraak, F.: A computationally efficient approximation of Dempster-Shafer theory. Int. Journal of Man-Machine Studies 30, 525–536 (1989)

The DSm Approach as a Special Case of the Dempster-Shafer Theory*

Milan Daniel

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, CZ - 182 07, Prague 8, Czech Republic
milan.daniel@cs.cas.cz

Abstract. This contribution deals with a belief processing which enables managing of multiple and overlapping elements of a frame of discernment. An outline of the Dempster-Shafer theory for such cases is presented, including several types of constraints for simplification of its large computational complexity. DSmT — a new theory rapidly developing the last five years — is briefly introduced. Finally, it is shown that the DSmT is a special case of the general Dempster-Shafer approach.

Keywords: Belief functions, Dempster-Shafer theory, DSm theory, Constraints, Overlapping elements, Exclusive elements, Non-separable elements.

1 Introduction

Belief functions are one of the widely used formalisms for uncertainty representation and processing that enable representation of incomplete and uncertain knowledge, belief updating and combination of evidence. They were originally introduced as a principal notion of the Dempster-Shafer Theory (DST) or the Mathematical Theory of Evidence [6].

For a combination of beliefs, Dempster's rule of combination is used in DST. Since the Dempster-Shafer theory publication, a series of modifications of Dempster's rule were suggested and alternative approaches were created. The classical ones are Dubois-Prade's rule [4] and Yager's rule of belief combination [10].

A new approach is the Dezert-Smarandache (or Dempster-Shafer modified) theory (DSmT) with its DSm rule of combination. There are two main differences: 1) mutual exclusivity of elements of a frame of discernment is not assumed in general; mathematically, this means that belief functions are not defined on the power set of the frame but on a so-called hyper-power set, i.e., on the Dedekind lattice defined by the frame; 2) a new combination mechanism which overcomes the problems with conflict among the combined beliefs and which also enables a dynamic fusion of beliefs.

* This work was supported by the grant 1ET100300419 GA AV ČR, and in part by the Institutional Research Plan AV0Z10300504 "Computer Science for the Information Society: Models, Algorithms, Applications".

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 381–392, 2007.

The classical Shafer's frame of discernment with exclusive elements is considered the special case of a so-called hybrid DSm model.

On the other hand, as it is presented in this study, the DSm approach is a special case of the Dempster-Shafer theory working on a frame of discernment with overlapping elements. The same holds true for both the basic DSm free model and any hybrid DSm models. To prove this, an outline of the Dempster-Shafer theory on frames with overlapping elements is formalized in this contribution.

Useful preliminaries are presented in Sect. 2. The third section presents a general outline of the Dempster-Shafer theory on a frame of discernment with overlapping and multiple elements. In Sect. 4, an important emphasis is devoted to constraints, which are used for better specification of the types of belief functions used in a given application area and for necessary decrease of a large computational complexity to a level comparable with the complexity of DSmT. A special focus is devoted to non-separable elements of a frame, which are crucial for relation with DSmT, see Sect. 4.3. A series of problems for future research are opened throughout the entire Sect. 4. The subsequent section briefly recalls DSmT, and Sect. 6 presents a special case of the Dempster-Shafer theory which is equivalent to DSmT, including subcases equivalent to particular hybrid models of DSmT.

2 Preliminaries

Let us assume an exhaustive finite *frame of discernment* $\Omega = \{\omega_1, ..., \omega_n\}$, whose elements are mutually exclusive.

A *basic belief assignment (bba)* is a mapping $m : \mathcal{P}(\Omega) \longrightarrow [0, 1]$, such that $\sum_{A \subseteq \Omega} m(A) = 1$, the values of bba are called *basic belief masses (bbm)*[1]. A *belief function (BF)* is a mapping $Bel : \mathcal{P}(\Omega) \longrightarrow [0, 1]$, $Bel(A) = \sum_{\emptyset \neq X \subseteq A} m(X)$, belief function Bel uniquely corresponds to bba m and vice-versa.

$\mathcal{P}(\Omega) = \{X | X \subseteq \Omega\}$ is often denoted also by 2^Ω. Let us define also a *restricted power set* as $\mathcal{P}(\Omega|X) = \{Y | X \subseteq Y \subseteq \Omega\}$, there is $\mathcal{P}(\Omega|\emptyset) = \mathcal{P}(\Omega)$ and $\mathcal{P}(\Omega|\Omega) = \{\Omega\}$. A *Borel field* (or σ*-field*) on Ω is a subset $\mathcal{B} \subseteq \mathcal{P}(\Omega)$ such that $\emptyset \in \mathcal{B}$, $\Omega \in \mathcal{B}$, $\overline{A} \in \mathcal{B}$ for all $A \in \mathcal{B}$, and $A \cup B \in \mathcal{B}$ for all $A, B \in \mathcal{B}$.

Dempster's (conjunctive) rule of combination $\oplus$ is given as $(m_1 \oplus m_2)(A) = \sum_{X \cap Y = A} K m_1(X) m_2(Y)$ for $A \neq \emptyset$, where $K = \frac{1}{1-\kappa}$, $\kappa = \sum_{X \cap Y = \emptyset} m_1(X) m_2(Y)$, and $(m_1 \oplus m_2)(\emptyset) = 0$, see [6]; putting $K = 1$ and $(m_1 \oplus m_2)(\emptyset) = \kappa$ we obtain the *non-normalized conjunctive rule of combination* $\textcircled{\cap}$, see e. g. [9].

Yager's rule of combination $\textcircled{Y}$, see [10], is given as $(m_1 \textcircled{Y} m_2)(\emptyset) = 0$, $(m_1 \textcircled{Y} m_2)(A) = \sum_{X,Y \subseteq \Theta,\, X \cap Y = A} m_1(X) m_2(Y)$ for $\emptyset \neq A \subset \Theta$, and $(m_1 \textcircled{Y} m_2)(\Theta) = m_1(\Theta) m_2(\Theta) + \sum_{X,Y \subseteq \Theta,\, X \cap Y = \emptyset} m_1(X) m_2(Y)$;

Dubois-Prade's rule of combination $\textcircled{DP}$ is given as $(m_1 \textcircled{DP} m_2)(A) = \sum_{X,Y \subseteq \Theta,\, X \cap Y = A} m_1(X) m_2(Y) + \sum_{X,Y \subseteq \Theta,\, X \cap Y = \emptyset, X \cup Y = A} m_1(X) m_2(Y)$ for $\emptyset \neq A \subseteq \Theta$, and $(m_1 \textcircled{DP} m_2)(\emptyset) = 0$, see [4].

[1] $m(\emptyset) = 0$ is often assumed in accordance with Shafer's definition [6]. A classical counter example is Smets' Transferable Belief Model (TBM) which admits $m(\emptyset) \geq 0$.

3 Dempster-Shafer Theory on Frame of Discernment with Overlapping or Multiple Elements

Let us assume an exhaustive finite *frame of discernment* $\Omega = \{\omega_1, ..., \omega_n\}$, whose elements are not mutually exclusive now. The elements of the frame can be overlapping as in the motivation example of DSmT (see Sect. 5.1) and in applications of DSmT (see the second parts of each of the volumes [7,8]), or several of them can be true at a time, thus we work with k-tuples of elements and we search for the right k-tuple, we work with belief about k-tuples and sets of k-tuples[2].

We can use a power set $\mathcal{P}(\Omega)$ (or $\mathcal{P}(\Omega) \setminus \emptyset$ to be more precise) instead of Ω and we apply the Dempster-Shafer approach to it. Hence we obtain a basic belief assignment as a mapping $m : \mathcal{P}(\mathcal{P}(\Omega)) \longrightarrow [0, 1]$, such that $\sum_{A \subseteq \mathcal{P}(\Omega)} m(A) = 1$, and a belief function as a mapping $Bel : \mathcal{P}(\mathcal{P}(\Omega)) \longrightarrow [0, 1]$, $Bel(A) = \sum_{\emptyset \neq X \subseteq A} m(X)$, for any $A \in \mathcal{P}(\mathcal{P}(\Omega))$, which represent our belief on $\mathcal{P}(\mathcal{P}(\Omega))$.

To express a belief about a subset A of the original frame Ω (i.e., a belief that some element of A is in the true k-tuple), we have to sum our belief over sets of k-tuples intersecting A. Hence we obtain $Bel'(A) = \sum_{\{Y \in X | Y \cap A \neq \emptyset\} = X} m(X) = \sum_{(\forall Y \in X)(Y \cap A \neq \emptyset)} m(X) = Bel(\bigcup_{\omega_i \in A} \mathcal{P}(\Omega|\{\omega_i\}))$ for $A \subseteq \Omega$; specially $Bel'(\{\omega_a\}) = Bel(\mathcal{P}(\Omega|\{\omega_a\}))$ for $\omega_a \in \Omega$.

In our approach, we have to distinguish whether a belief mass is assigned to an element ω_i of the frame of discernment (or to a set of elements $\{\omega_{j_1}, ..., \omega_{j_k}\}$) exclusively or non-exclusively, in our approach; i.e., we have to distinguish whether the belief mass is assigned just to 1-tuple (ω_i) only, or whether all k-tuples including ω_i are considered, or some set of k-tuples including ω_i is in question.

Example 1. Let suppose a 3-element frame of discernment $\Omega = \{a, b, c\}$.
(i) Let two believers (agents) assign belief masses non-exclusively to elements a and b respectively. Thus one assigns his belief mass to the set of all k-tuples containing element a: $A = \{(a), (a, b), (a, c), (a, b, c)\} = \mathcal{P}(\Omega|\{a\})$, the second analogically assigns his belief mass to the set of all k-tuples containing element b: $B = \{(b), (a, b), (b, c), (a, b, c)\} = \mathcal{P}(\Omega|\{b\})$. When combining their beliefs using Dempster's rule (or its alternatives mentioned above), we have to assign the resulting belief mass to intersection of the sets $A \cap B = \{(a, b), (a, b, c)\} = \mathcal{P}(\Omega|\{a, b\})$, i.e. to the set of all k-tuples containing both a and b.
(ii) If our believers assign their belief masses to the same elements exclusively, we obtain $A^x = \{(a)\}$, $B^x = \{(b)\}$, and $A^x \cap B^x = \emptyset$, as both beliefs are in contradiction, hence the belief mass should be normalized, assigned to union or to whole Ω, according to the combination rule which is used.
(iii) Let our believers assign their belief masses non-exclusively to elements a or b and a or c respectively. Thus their belief masses are assigned to the set of all k-tuples containing element a or b: $AB = \{(a), (b), (a, b), (a, c), (b, c), (a, b, c)\}$ and

[2] All k-tuples which we speak about in this paper have non-repeating members. They are just sets. There is neither ordering nor repeating of their elements. We use brackets to underline that elements of k-tuple appear simultaneously together, to distinguish it from a set of tuples where we know that just one of them appears (just one of them is right).

to the set of all k-tuples containing element a or c: $AC = \{(a), (c), (a, b), (a, c), (b, c), (a, b, c)\}$. Within the combination the resulting belief is assigned to intersection $AB \cap AC = \{(a), (a, b), (a, c), (b, c), (a, b, c)\}$.
(iv) In the fully exclusive version of the previous case, we have $A^x B^x = \{(a), (b)\}$ and $A^x C^x = \{(a), (c)\}$ with intersection $A^x B^x \cap A^x C^x = \{(a)\}$.
(v) Let the first assign his belief mass non-exclusively to a and exclusively to b, and the second exclusively to a or non-exclusively to c. Hence we obtain sets $AB^x = \{(a), (b), (a, b), (a, c), (a, b, c)\}$ and $A^x C = \{(a), (c), (a, c), (b, c), (a, b, c)\}$ with intersection $AB^x \cap A^x C = \{(a), (a, c), (a, b, c)\}$.
(vi) And there is a series of other possibilities as as basic belief assignment is defined on $\mathcal{P}(\mathcal{P}(\Omega))$.

In a special case, all belief masses are assigned to all elements and their sets exclusively by all believers, thus we have just sets of exclusive elements $\{(a)\}, \{(b)\}, \{(c)\}, \{(a), (b)\}, \{(a), (c)\}, \{(b), (c)\}, \{(a), (b), (c)\}$, i.e., we work on $\mathcal{P}(\{(a), (b), (c)\})$ when assuming the frame from Example 1 above. In the same way it holds for any finite frame of discernment, when all belief masses are assigned exclusively to single elements (1-tuples) (ω_i) and sets of them, i.e., we work on $\mathcal{P}(\{(\omega_1), ..., (\omega_n)\})$ in such a case. Hence we have shown that the following fact holds:

Fact 1. *The Dempster-Shafer approach to belief on a classical exclusive frame of discernment is a special case of our generalized approach working with overlapping or multiple elements.*

3.1 Decreasing of Computational Complexity

When considering a full system of belief functions defined on the "double" power set $\mathcal{P}(\mathcal{P}(\Omega))$ for a finite frame of discernment $\Omega = \{\omega_1, ..., \omega_n\}$, a computational complexity corresponds to the size of $\mathcal{P}(\mathcal{P}(\Omega))$, i.e. to 2^{2^n}, thus we obtain possibly $16, 256, 65\,536, ...$ different belief masses for $n = 2, 3, 4, ...$.

Even if we do not accept positive belief masses and beliefs for empty set and empty tuples (0-tuples) in accordance with the classis Dempster-Shafer approach, i.e., positive belief masses are accepted only on $\mathcal{P}(\mathcal{P}(\Omega) \setminus \emptyset) \setminus \emptyset$, we can have still $2^{2^n - 1} - 1$ positive values, thus we obtain $7, 127, 32\,767, ...$ values for $n = 2, 3, 4, ...$. Hence general belief functions are hardy computable even for $n = 5$, as we have the original complexity only divided by two.

Nevertheless, we do not need all combinatorically possible n-tuples and all their sets in any application, thus we can use some constraints for their generation. Hence the computational complexity can be significantly decreased almost down to computational complexity of the classical Dempster-Shafer approach, i.e. to $\mathcal{O}(2^n)$, because the classical Dempster-Shafer approach is a special case as we have shown above. A restriction of definition domain of belief assignment and belief functions using constraints corresponds to the assumption that they can be defined on a more general form of domain, on Borel fields (σ-fields), [5].

4 Constraints for Definition of Belief Assignments

We have seen that to decrease a computational complexity it is necessary to make as many restrictions of domain as possible with respect to a processed application to decrease a computational complexity. A simple way is non-distinguishing of some elements, i.e. a coarsening of the original frame of discernment Ω; this simply decreases n itself.

Another possibility is acceptation of exclusivity of some elements, i.e. admission of possibility of overlapping only for some elements of the frame. Thus we have a kind of coarsening on $\mathcal{P}(\Omega)$ this time.

The third type of constraint is assuming that some class of sets (k-tuples) is forced to have zero belief mass, similarly, as is assumed for the empty set in the classic Dempster-Shafer approach.

4.1 Non-distinguishable Elements Coarsening of Frame of Discernment Ω

Let some elements of a frame of discernment be mutually non-distinguishable or mutually equivalent in an application in question. Thus we have an equivalence relation $\mathcal{E}$ on the frame of discernment Ω and we can make a coarsening of Ω according to it. It is the same as if we defined basic belief assignment on the "double" power set of classes of equivalence $\mathcal{E}$ instead of the original Ω. Hence $n = |\Omega|$ is decreased to $m = |\mathcal{E}|$, from cardinality of Ω to number of equivalence classes of relation $\mathcal{E}$.

This simple case can really decrease computational complexity of real applications, but it is not interesting from a theoretical point of view.

4.2 Exclusive and Non-exclusive Elements Coarsening of Power Set of $\mathcal{P}(\Omega)$

Let us assume that some elements of the frame of discernment Ω are exclusive now. Let us assume, e.g., 3-element frame $\Omega = \{a, b, c\}$, where a is mutually exclusive with b, we can denote it $Excl(a, b)$. a and b cannot appear at a time, thus both $\{(a, b)\}$ and $\{(a, b, c)\}$ are equivalent to the empty set, i.e. $\{(a, b)\} \equiv \{(a, b, c)\} \equiv \emptyset$, and their belief masses must be equal to 0. There remain 5 possible k-tuples $(a), (b), (c), (a, c), (b, c)$, and belief assignment is defined on the power set of 5-element set $\{(a), (b), (c), (a, c), (b, c)\}$. Hence we have $2^5 - 1$ subsets, i.e. at most 31 positive values of basic belief masses instead of original $2^7 - 1 = 127$.

We have the following equivalence on "double" power set $\mathcal{P}(\mathcal{P}(\Omega))$: $\emptyset \equiv \{(a, b)\} \equiv \{(a, b, c)\} \equiv \{(a, b), (a, b, c)\}$, $\{(a)\} \equiv \{(a), (a, b)\} \equiv \{(a), (a, b, c)\} \equiv \{(a), (a, b), (a, b, c)\}$, $\{(a), (a, c)\} \equiv \{(a), (a, b), (a, c)\} \equiv \{(a), (a, c), (a, b, c)\} \equiv \{(a), (a, b), (a, c), (a, b, c)\}$, etc. ...

If a is moreover mutually exclusive with c, i.e., if $\{(a, b)\} \equiv \{(a, b, c)\} \equiv \emptyset \equiv \{(a, c)\}$, we have at most $2^4 - 1 = 15$ positive values of basic belief masses.

Let us, now, assume a special case of general finite frame of discernment $\Omega = \{\omega_1, ..., \omega_n\}$, where only ω_1 and ω_2 are not exclusive and all the other

elements are mutually exclusive each other and also mutually exclusive with ω_1 and ω_2. There is $n+1$ tuples $(\omega_1, \omega_2), (\omega_1), (\omega_2), ..., (\omega_n)$, which are not equivalent to $\emptyset$, whereas all the other tuples are equivalent to empty set. Hence we have at most $2^{n+1} - 1 \sim 2(2^n - 1)$ positive belief masses for set of tuples. Thus the computational complexity is only roughly twice higher in comparison with the classical Dempster-Shafer approach using only exclusive elements.

In general, more complex cases of exclusivity constraints can appear, e.g. $Excl(a, b, c) \& Excl(d, e)$ on $\Omega = \{a, b, c, d, e\}$, where elements a, b, c are all mutually exclusive and other two d, e are again mutually exclusive, i.e. $\{(a, b)\} \equiv \{(a, c)\} \equiv \{(b, c)\} \equiv \{(a, b, c)\} \equiv \{(d, e)\} \equiv \emptyset$, i.e. also $\{(a, b, ...)\} \equiv \{(a, c, ...)\} \equiv \{(b, c, ...)\} \equiv \{(a, b, c, ...)\} \equiv \{(d, e, ...)\} \equiv \emptyset$.

All the above coarsening have a common feature that some subclass of $\mathcal{P}(\Omega)$ is equivalent to the empty set, and all other subsets of Ω (i.e. k-tuples) are mutually non-equivalent. It remains an open question whether it would be reasonable to consider also some more general equivalence on $\mathcal{P}(\Omega)$ as constraint for definition of belief assignments, e.g. the equivalence as it follows $\{(a, b)\} \equiv \{(a, c)\} \equiv \{(a, b, c)\} \equiv \emptyset$, $\{(b, c)\} \equiv \{(b)\} \not\equiv \emptyset$ (and $\{(a)\} \not\equiv \{(c)\} \not\equiv \{(b)\} \equiv \{(b, c)\} \not\equiv \{(a)\} \not\equiv \emptyset$, $\{(c)\} \not\equiv \emptyset$). Whether it is reasonable to consider exclusivity of a couple or generally k-tuple of elements of Ω, and similarly.

4.3 Non-separable Elements Forced Zero Belief Masses

Let us assume now that one element ω_a of the frame is not separable from the others [3], i.e., that it is not possible to assign any belief mass neither to single (ω_a) nor to any set of k-tuples containing ω_a separately from the other elements; i.e., that any belief mass in favour of ω_a must be assigned to the set of all k-tuples containing ω_a, and any belief mass in favour of the m-tuple $(\omega_a, \omega_{i_2}, ..., \omega_{i_m})$, where $2 \leq m \leq n-1$, must be assigned to all k-tuples containing $\{\omega_a, \omega_{i_2}, ..., \omega_{i_m}\}$ for all $m < k \leq n$.

Similarly, any belief mass in favour of set $\{(\omega_a), (\omega_{j_2}), ..., (\omega_{j_m})\}$, where $2 \leq m \leq n-1$, must be assigned to union of the set of all k-tuples containing ω_a with $\{(\omega_{j_2}), ..., (\omega_{j_m})\}$, and analogously for any belief mass in favour of the set of tuples including some tuple(s) with ω_a. All other sets of tuples containing at least one tuple containing ω_a are forced to have zero basic belief mass, they are *constrained to have zero basic belief mass*, or simply constrained (to zero).

Let us show it on the following example on a simple frame of discernment.

Example 2. Let us assume $\Omega = \{a, b, c\}$ again. Let a be not separable from b and c now. No believer can assign anything separately to single $\{(a)\}$ or to $\{(a), (a, b)\}$ as anything that is believed in favour of a must be assigned to set of all tuples including a, etc.

Thus $(a), (a, b), (a, c)$ and their sets are forced to have zero belief masses. What is believed in favour of a must be assigned to $\{(a), (a, b), (a, c), (a, b, c)\}$, and similarly for $\{(a), (a, b)\}$ and $\{(a), (a, c)\}$. What is believed in favour of $\{(a, b)\}$

[3] We need this for the restriction of the "double" power set to the definition domain of the DSmT, see Sections 5 and 6.

must be assigned to $\{(a,b),(a,b,c)\}$, what is believed in favour of $\{(a,c)\}$ must be assigned to $\{(a,c),(a,b,c)\}$, only the set $\{(a,b,c)\}$, which includes all the elements simultaneously, behaves as is usual. Thus all other proper subsets of $\{(a),(a,b),(a,c),(a,b,c)\}$ are forced to have zero belief masses, i.e. constrained (to zero). Hence all the following sets are also constrained: $\{(a)\}$, $\{(a),(a,b)\}$, $\{(a),(a,c)\}$, $\{(a),(a,b,c)\}$, $\{(a),(a,b),(a,c)\}$, $\{(a),(a,b),(a,b,c)\}$, $\{(a),(a,c),(a,b,c)\}$, $\{(a,b)\}$, $\{(a,c)\}$, and $\{(a,b),(a,c)\}$. We can count that there are 10 proper subsets of $\{(a),(a,b),(a,c),(a,b,c)\}$, which are constrained to zero.

Let us note a very important fact that neither $\{(a)\}$, $\{(a,b)\}$, $\{(a,c)\}$ nor any from the other seven constrained subsets of $\{(a),(a,b),(a,c),(a,b,c)\}$ must be equivalent to the empty set. These sets are non-empty in general, they only have zero belief masses. Otherwise, all five subsets which are allowed to take positive belief masses would be equivalent to $\{(a,b,c)\}$, which is not our case.

Let us take one of the above 10 sets and denote it as X, and let Y be a subset $(\mathcal{P}(\Omega)\setminus\emptyset)\setminus\{(a),(a,b),(a,c),(a,b,c)\}$, i.e., subset of $\{(b),(c),(b,c)\}$. Anything that is believed in favour of $X\cup Y$ must be assigned to union of Y with the entire set to what is assigned which is believed in favour of X, i.e. to $\{(a),(a,b),(a,c),(a,b,c)\}\cup Y$, $\{(a,b),(a,c),(a,b,c)\}\cup Y$, $\{(a,b),(a,b,c)\}\cup Y$ $\{(a,c),(a,b,c)\}\cup Y$, or to $\{(a,b,c)\}\cup Y$, respectively. As there are 7 non-empty subsets of $\{(b),(c),(b,c)\}$, i.e. 7 options for the selection of Y, there is in combination with 10 options for X other 70 subsets of $\mathcal{P}(\Omega)$, which are constrained to have zero belief mass; thus together 80 sets of k-tuples are constrained to zero.

E.g., what is believed in favour of $\{(a),(b)\}=\{(a)\}\cup\{(b)\}$ must be assigned to $\{(a),(a,b),(a,c),(a,b,c)\}\cup\{b\}=\{(a),(b),(a,b),(a,c),(a,b,c)\}$ and $\{(a),(b)\}$ must have zero belief mass, which is believed in favour of $\{(b),(a,b),(a,c)\}=\{(a,b),(a,c)\}\cup\{b\}$ must be assigned to $\{(a,b),(a,c),(a,b,c)\}\cup\{b\}=\{(b),(a,b),(a,c),(a,b,c)\}$ and $\{(a,b),(a,c),(a,b,c)\}$ must have zero belief mass, and similarly.

The number of non-empty sets of tuples on our 3-element frame of discernment was decreased from $2^{2^3-1}-1=127$, which have possibly positive belief mass in a general case, to $127-80=47$, which are allowed to obtain positive belief mass in our example. The more elements is non-separable, the more is decreased the number of sets of tuples, which are allowed to obtain positive believe masses.

Let us turn our attention to a general finite case, where ω_a is non-separable from the other elements again. We can use a restricted power set for description of sets of tuples containing ω_a as $\mathcal{P}(\Omega|\{\omega_a\})$, similarly to the above example, some of its elements (sets of tuples) are forced to have a zero belief mass, the others are not. We can observe that sets of tuples $\mathcal{A}(\{\omega_a\})$, which are *allowed to take positive belief masses* (when ω_a is non-separable), are equal to some restricted power set of the frame or more generally to a union of some restricted power sets. Trivially, the set of all tuples containing ω_a is allowed, thus $\mathcal{P}(\Omega|\{\omega_a\})\in\mathcal{A}(\{\omega_a\})$. For more general $X\subseteq\mathcal{P}(\Omega|\{\omega_a\})$ it holds that $X\in\mathcal{A}(\{\omega_a\})$ *iff* $(\exists(\omega_{a_{11}},...,\omega_{a_{1k_1}}),...,(\omega_{a_{m1}},...,\omega_{a_{mk_m}}))(X=\bigcup_{i=1}^{m}\mathcal{P}(\Omega|\{\omega_{a_{i1}},...,\omega_{a_{ik_i}}\}))$.

This can be demonstrated by the previous Example 2, where the following holds true: $\{(a),(a,b),(a,c),(a,b,c)\}=\mathcal{P}(\{a,b,c\}|\{a\})$, $\{(a,b),(a,b,c)\}=$

$\mathcal{P}(\{a,b,c\}|\{a,b\})$, $\{(a,c),(a,b,c)\} = \mathcal{P}(\{a,b,c\}|\{a,c\})$, $\{(a,b),(a,c),(a,b,c)\} = \mathcal{P}(\{a,b,c\}|\{a,b\}) \cup \mathcal{P}(\{a,b,c\}|\{a,c\})$, and $\{(a,b,c)\} = \mathcal{P}(\{a,b,c\}|\{a,b,c\})$. All other subsets of $\mathcal{P}(\{a,b,c\}|\{a\})$ are constrained to zero, see above.

Let us denote the set of all subsets of $\mathcal{P}(\Omega|\{\omega_a\})$ which are in $\mathcal{A}(\{\omega_a\})$ as $\mathcal{A}_0(\{\omega_a\})$. For completely general $X \subseteq \mathcal{P}(\Omega)$ it holds that
$X \in \mathcal{A}(\{\omega_a\})$ *iff* $(\exists X_0, X_1)(X = X_0 \cup X_1, X_0 \in \mathcal{A}_0(\{\omega_a\}), X_1 \subseteq \mathcal{P}(\Omega) \setminus \mathcal{P}(\Omega|\{\omega_a\}))$.

This can be demonstrated by the previous Example 2 again, where have the following: $\{(b),(a,b),(a,c),(a,b,c)\} = (\mathcal{P}(\{a,b,c\}|\{a,b\}) \cup \mathcal{P}(\{a,b,c\}|\{a,c\})) \cup \{(b)\}$, $\{(b),(a,b),(b,c),(a,b,c)\} = \mathcal{P}(\{a,b,c\}|\{a,b\}) \cup \{(b),(b,c)\}$, and similarly.

We have to underline a principal difference between constraints defined by exclusive elements or non-distinguishable elements and just presented constraint defined by non-separable elements. The previous two classes of constraints decrease a size of a definition domain of generalized basic belief assignments and belief functions, which are defined on the power sets (of k-tuples) or on their subsets, which are Borel fields in general. Whereas the present type of constraints keeps the original power set (or Borel field in full generality) as the definition domain, but some parts of the definiton domains are forced to obtain always zero belief masses. The first two cases reducing a structure as classes of a constraining equivalence $\mathcal{E}$ on $\mathcal{P}(\mathcal{P}(\Omega))$ are used instead of the whole $\mathcal{P}(\mathcal{P}(\Omega))$. In the present case the whole structure is kept; nevertheless, some of its parts are constrained to zero, hence a computational complexity is decreased regardless of keeping the structure.

To finish this subsection, it remains to show that class $\mathcal{A}(\{\omega_a\})$ of the subsets of $\mathcal{P}(\Omega)$ which are allowed to have positive belief masses, is closed with respect to classic rules, especially Dempster's rule, i.e., that combination is really performed in the non-constrained part of a corresponding definition domain.

Idea of proof: We have to show that class $\mathcal{A}(\{\omega_a\})$ is closed with respect to combination. I.e., that no positive belief mass is assigned to any set out of $\mathcal{A}(\{\omega_a\})$ within the belief combination. All the classic combination rules, Dempster's rule, Yager's rule and Dubois-Prade rule computes values of resulting basic belief masses using multiples of original values. If one of two input basic belief masses is equal zero, then also their multiple is zero, and there is no problem there. Thus it is sufficient to focus our attention only to multiples of two positive values, i.e. values assigned to sets X and Y both from $\mathcal{A}(\{\omega_a\})$. The corresponding multiple is assigned to intersection $X \cap Y$ if it is non-empty, otherwise it is normalized by Dempster's rule, assigned to the whole frame by Yager's rule, or assigned to union $X \cup Y$ by Dubois-Prade rule. $\mathcal{P}(\Omega) = \mathcal{P}(\Omega|\{\omega_a\}) \cup (\mathcal{P}(\Omega) \setminus \mathcal{P}(\Omega|\{\omega_a\})) \in \mathcal{A}(\{\omega_a\})$. Thus it is enough to show that $\mathcal{A}(\{\omega_a\})$ is closed w.r.t. operations of intersection and union.

$X \cap Y = (\bigcup_{i=1}^{m} \mathcal{P}(\Omega|\{\omega_{a_{i1}}, ..., \omega_{a_{ik_i}}\}) \cup X_1) \cap (\bigcup_{j=1}^{r} \mathcal{P}(\Omega|\{\omega_{a_{j1}}, ..., \omega_{a_{jk_j}}\}) \cup Y_1) = \bigcup_{i,j=1}^{m,r} (\mathcal{P}(\Omega|\{\omega_{a_{i1}} ..., \omega_{a_{ik_i}}\}) \cap \mathcal{P}(\Omega|\{\omega_{a_{j1}}, ..., \omega_{a_{jk_j}}\}) \cup (\bigcup_{i=1}^{m} \mathcal{P}(\Omega|\{\omega_{a_{i1}}, ..., \omega_{a_{ik_i}}\}) \cap Y_1) \cup (\bigcup_{j=1}^{r} \mathcal{P}(\Omega|\{\omega_{a_{j1}}, ..., \omega_{a_{jk_j}}\}) \cap X_1) \cup (X_1 \cap Y_1) = \bigcup_{i,j=1}^{m,r} (\mathcal{P}(\Omega|\{\omega_{a_{i1}} ..., \omega_{a_{ik_i}}\} \cup \{\omega_{a_{j1}}, ..., \omega_{a_{jk_j}}\}) \cup (X_1 \cap Y_1) \in \mathcal{A}(\{\omega_a\})$.

$X \cup Y = (\bigcup_{i=1}^{m} \mathcal{P}(\Omega|\{\omega_{a_{i1}}, ..., \omega_{a_{ik_i}}\}) \cup X_1) \cup (\bigcup_{j=1}^{r} \mathcal{P}(\Omega|\{\omega_{a_{j1}}, ..., \omega_{a_{jk_j}}\}) \cup Y_1) =$
$(\bigcup_{i=1}^{m} \mathcal{P}(\Omega|\{\omega_{a_{i1}} ..., \omega_{a_{ik_i}}\}) \cup \bigcup_{j=1}^{r} \mathcal{P}(\Omega|\{\omega_{a_{j1}}, ..., \omega_{a_{jk_j}}\})) \cup (X_1 \cup Y_1) \in \mathcal{A}(\{\omega_a\})$.
□

Similarly to the previous subsection, more general constraining conditions using non-separable elements should be discussed and investigated in future.

4.4 Overview of Constraints

Let us make a brief overview of possible types of constraints to summarize this section. There can be the following types of constraints to better specify a type of used belief functions and to decrease computational complexity:

- coarsening of the original frame of discernment Ω, i.e., an equivalence on the frame, belief assignment and belief functions are defined on the "double" power set of a coarsened frame of discernment,
- exclusive elements — coarsening of $\mathcal{P}(\Omega)$, i.e. an equivalence on $\mathcal{P}(\Omega)$:
  - mutual exclusivity of an element with all the other elements of the frame,
  - mutual exclusivity of an element with a group of elements of the frame,
  - mutual exclusivity of groups of elements of the frame,
- more general coarsening of $\mathcal{P}(\Omega)$, i.e. still the power set of equivalence classes,
- subsets of "double" power sets, i.e. more general Borel fields on $\mathcal{P}(\Omega)$,
- non-separable elements:
  - elements non-separable from all the other elements of the frame Ω,
  - elements non-separable from a subset of Ω,
  - non-separability of groups of elements of the frame,

  parts of a definition domain (i.e. parts of the power set or of Borel field in general) are constrained to zero,
- a combination of previous constraints.

We can see that only several simple classes of constraint have been discussed here, namely such that are necessary for solving the tasks of DSmT, and that there is a wide area still open for future investigation.

5 A Brief Introduction to DSm Theory

The DSm theory (DSmT) is a new theory which appeared five years ago in 2002 [3], and which is in permanent dynamic evolution, see [7,8], and the anouncement of a new volume. For the brand new advances of DSmT we can see its homepage.

5.1 Dedekind Lattice, Basic DSm Notions

Dempster-Shafer modified Theory or Dezert-Smarandache Theory (DSmT) by Dezert and Smarandache [3,7] allows mutually overlapping elements of a frame of discernment. Thus, a frame of discernment is a finite exhaustive set of elements $\Omega = \{\omega_1, \omega_2, ..., \omega_n\}$, but not necessarily exclusive in DSmT. As an example, we can introduce a three-element set of colours $\{Red, Green, Blue\}$ from the DSmT

homepage[4]: DSmT allows that an object can have 2 or 3 colours at the same time: e.g. it can be both red and blue, or red and green and blue in the same time, it corresponds to a composition of the colours from the 3 basic ones.

DSmT uses basic belief assignments and belief functions defined analogically to the classic Dempster-Shafer theory (DST), but they are defined on a so-called hyper-power set or Dedekind lattice (which does not satisfy the conditions of Borel field), instead of the classic power set of the frame of discernment.

The *Dedekind lattice*, more frequently called *hyper-power set* D^Ω in DSmT, is defined as the set of all composite propositions built from elements of Ω with union and intersection operators $\cup$ and $\cap$ such that $\emptyset, \omega_1, \omega_2, ..., \omega_n \in D^\Omega$, and if $A, B \in D^\Omega$ then also $A \cup B \in D^\Omega$ and $A \cap B \in D^\Omega$, no other elements belong to D^Ω ($\omega_i \cap \omega_j \neq \emptyset$ in general, $\omega_i \cap \omega_j = \emptyset$ iff $\omega_i = \emptyset$ or $\omega_j = \emptyset$).

Thus the hyper-power set D^Ω of Ω is closed to $\cup$ and $\cap$ and $\omega_i \cap \omega_j \neq \emptyset$ in general, hence D^Ω is not a Borel field. Whereas the classic power set 2^Ω of Ω is closed to $\cup$, $\cap$ and complement, and $\omega_i \cap \omega_j = \emptyset$ for every $i \neq j$.

Examples of hyper-power sets. Let $\Omega = \{\omega_1, \omega_2\}$, we have $D^\Omega = \{\emptyset, \omega_1 \cap \omega_2, \omega_1, \omega_2, \omega_1 \cup \omega_2\}$, i.e., $|D^\Omega| = 5$. Let $\Omega = \{\omega_1, \omega_2, \omega_3\}$ now, we have $D^\Omega = \{\alpha_0, \alpha_1, ...\alpha_{18}\}$, where $\alpha_0 = \emptyset, \alpha_1 = \omega_1 \cap \omega_2 \cap \omega_3, \alpha_2 = \omega_1 \cap \omega_2, \alpha_3 = \omega_1 \cap \omega_3, ..., \alpha_{17} = \omega_2 \cup \omega_3, \alpha_{18} = \omega_1 \cup \omega_2 \cup \omega_3$, i.e., $|D^\Omega| = 19$ for $|\Omega| = 3$.

5.2 DSm Models

If we assume a Dedekind lattice (hyper-power set) according to the above definition without any other assumptions, i.e., all elements of an exhaustive frame of discernment can mutually overlap themselves, we refer to the *free DSm model* $\mathcal{M}^f(\Omega)$, i.e. about the DSm model free of constraints. In general, it is possible to add exclusivity or non-existential constraints into DSm models, we speak about *hybrid DSm models* in such cases.

An exclusivity constraint $\omega_1 \cap \omega_2 \overset{\mathcal{M}_1}{\equiv} \emptyset$ says that elements ω_1 and ω_2 are mutually exclusive in model $\mathcal{M}_1$, whereas both of them can overlap with ω_3. If we assume exclusivity constraints $\omega_1 \cap \omega_2 \overset{\mathcal{M}_2}{\equiv} \emptyset$, $\omega_1 \cap \omega_3 \overset{\mathcal{M}_2}{\equiv} \emptyset$, $\omega_2 \cap \omega_3 \overset{\mathcal{M}_2}{\equiv} \emptyset$, another exclusivity constraint directly follows them: $\omega_1 \cap \omega_2 \cap \omega_3 \overset{\mathcal{M}_2}{\equiv} \emptyset$. In this case all the elements of the 3-element frame of discernment $\Omega = \{\omega_1, \omega_2, \omega_3\}$ are mutually exclusive as in the classic Dempster-Shafer theory, and we call such hybrid DSm model as *Shafer's model* $\mathcal{M}^0(\Omega)$.

A non-existential constraint $\omega_3 \overset{\mathcal{M}_3}{\equiv} \emptyset$ brings a piece of additional information about a frame of discernment saying that ω_3 is impossible; it forces all the belief masses of $X \subseteq \omega_3$ to be equal to zero for any basic belief assignment in model $\mathcal{M}_3$. It represents a sure meta-information with respect to generalized belief combination which is used in a dynamic fusion.

5.3 The DSm Rules of Combination

There is a series of combination rules in DSmT. Originally in [3], only the DSm classic and the hybrid DSm rules were considered, i.e. the generalized conjunctive

[4] www.gallup.unm.edu/∼smarandache/DSmT.htm

rule and a slightly extended generalization of the Dubois-Prade rule, in fact, for detail see [2]. For a possibility of better comparison of the DSmT with the classic Dempster-Shafer theory also Dempster's and Yager's rules were generalized to the DSm hyper-power sets in [2]. Besides some new rules were defined in [8].

6 Dempster-Shafer Theory on a Frame of Discernment with Overlapping and Fully Non-separable Elements

6.1 Definition Domain of Belief Functions

Let us assume the Dempster-Shafer theory with overlapping elements of a frame of discernment in the sense of Sect. 3. Let us more assume that all the elements of the frame of discernment are non-separable from the others in the sense of Sect. 4.3. Belief assignments and belief functions are defined on $\mathcal{P}(\mathcal{P}(\Omega))$, but a lot of the subsets of $\mathcal{P}(\Omega)$ are constrained to have zero belief masses.

We have shown that the set of non-constrained elements is closed under $\cap$ and $\cup$ in Sect. 4.3. The set of all k-tuples containing ω_i, i.e. $\mathcal{P}(\Omega|\{\omega_i\})$, is not constrained for any ω_i, as we assume only non-separability constraints. Such a set corresponds to a belief assigned in favour of ω_i, for any $\omega_i \in \Omega$, and positive belief masses are allowed for all unions and intersections iteratively applied to these sets. Thus hyper-power set D^Ω generated by Ω is included in our constrained $\mathcal{P}(\mathcal{P}(\Omega))$. Moreover, hyper-power set D^Ω is equivalent to the constrained $\mathcal{P}(\mathcal{P}(\Omega))$ as all non-constrained proper subsets of $\mathcal{P}(\Omega|\{\omega_i\})$ are in a form $\mathcal{P}(\Omega|\{\omega_i\} \cup X) = \mathcal{P}(\Omega|\{\omega_i\}) \cap \mathcal{P}(\Omega|X)$ for some $X \subset \Omega$, in our case. Hence the $\mathcal{P}(\mathcal{P}(\Omega))$ constrained by full non-separability of all elements of Ω is equivalent to the free DSm model $\mathcal{M}^f(\Omega)$, where an element $\{(\omega_{11}, ..., \omega_{1k_1}), ..., (\omega_{j1}, ..., \omega_{jk_j})\}$ of $\mathcal{P}(\mathcal{P}(\Omega))$ fully corresponds to the disjunctive normal form of an element $(\omega_{11} \cap ... \cap \omega_{1k_1}) \cup ... \cup (\omega_{j1} \cap ... \cap \omega_{jk_j})$ of $\mathcal{M}^f(\Omega)$.

As we can combine together the constraints given by non-separable elements and the exclusivity constraints given by exclusive elements of Ω, we can add exclusivity constraints in the same way as in DSmT. And a strange DSm non-existence constraint is equivalent to exclusivity of an element ω_i with whole Ω including ω_i itself, i.e., all k-tuples including ω_i are equivalent to the empty set for $k \geq 1$ (including (ω_i)), hence they are constrained. Thus for any hybrid DSm model $\mathcal{M}(\Omega)$, there is a set of constraints such that constrained $\mathcal{P}(\mathcal{P}(\Omega))$ is equivalent to $\mathcal{M}(\Omega)$. Hence we have just proved the following theorem.

Theorem 1. *Any DSm model $\mathcal{M}(\Omega)$ is a special case of the constrained double power set $\mathcal{P}(\mathcal{P}(\Omega))$ from the Dempster-Shafer approach to overlapping elements.*

6.2 Several Notes to Belief Combination

We have shown above that DSmT is a special case of Demspter-Shafer theory from the perspective of their respective definition domains. Thus, also all combination rules from DSmT can be used and studied from the point of view of the Dempster-Shafer theory. Trivially, Dempster's and Yager's rules which have

been generalized from the Dempster-Shafer theory on frames of discernment with exclusive elements to DSmT are common to the both of these approaches.

An interesting open problem for future is to investigate the full relation of the hybrid DSm rule and of the Dubois-Prade rule on the frame of discernment with overlapping elements including different types of the constrained "double" power sets. Also analogically, relation of a family of new DSm PCR rules to both the full and constrained "double" power sets.

7 Conclusion

DSmT brings the new domain of belief functions with a lot of variants via various hybrid DMs models. We have proved that any DSm hybrid model is a special case of constrained "double" power set $\mathcal{P}(\mathcal{P}(\Omega))$ of the frame of discernment used in the Dempster-Shafer theory for frames of discernment with overlapping elements. We have shown that the DSm belief combination can be also interpreted from the point of view of the Dempster-Shafer theory.

This enables us to cover the entire DSmT using the approach of the Dempster-Shafer theory, and it also allows for a better placement of the DSmT among the other approaches to the belief functions.

To achieve this, we have suggested several types of constraints in the Dempster-Shafer theory on overlapping and multiple elements. We have formalized basic types of these constraints in this paper, and we have opened a series of new questions for future research here.

References

1. Daniel, M.: Distribution of Contradictive Belief Masses in Combination of Belief Functions. In: Bouchon-Meunier, B., Yager, R.R., Zadeh, L.A. (eds.) Information, Uncertainty and Fusion, pp. 431–446. Kluwer Academic Publishers, Dordrecht (2000)
2. Daniel, M.: A Generalization of the Classic Combination Rules to DSm Hyper-power Sets. Information & Security. An Internat. Journal 20, 50–64 (2006)
3. Dezert, J.: Foundations for a New Theory of Plausible and Paradoxical Reasoning. Information and Security, An International Journal 9 (2002)
4. Dubois, D., Prade, H.: Representation an combination of uncertainty with belief functions and possibility measures. Computational Intelligence 4, 244–264 (1988)
5. Klir, G.J., Folger, T.A.: Fuzzy Sets, Uncertainty, and Information. Prentice-Hall, Englewood Cliffs (1988)
6. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
7. Smarandache, F., Dezert, J.: Advances and Applications of DSmT for Information Fusion. American Research Press, Rehoboth (2004)
8. Smarandache, F., Dezert, J.: Advances and Applications of DSmT for Information Fusion. vol. 2, American Research Press, Rehoboth (2006)
9. Smets, Ph.: The combination of evidence in the transferable belief model. IEEE-Pattern analysis and Machine Intelligence 12, 447–458 (1990)
10. Yager, R.R.: On the Demspter-Shafer framework and new combination rules. Information Sciences 41, 93–138 (1987)

Interpreting Belief Functions as Dirichlet Distributions

Audun Jøsang[1] and Zied Elouedi[2]

[1] Information Security Institute
Queensland University of Technology
Brisbane, Australia
a.josang@qut.edu.au
[2] Tunisian Institute of Management
Tunisia
zied.elouedi@gmx.fr

Abstract. Traditional Dempster Shafer belief theory does not provide a simple method for judging the effect of statistical and probabilistic data on belief functions and vice versa. This puts belief theory in isolation from probability theory and hinders fertile cross-disciplinary developments, both from a theoretic and an application point of view. It can be shown that a bijective mapping exists between Dirichlet distributions and Dempster-Shafer belief functions, and the purpose of this paper is to describe this correspondence. This has three main advantages; belief based reasoning can be applied to statistical data, statistical and probabilistic analysis can be applied to belief functions, and it provides a basis for interpreting and visualizing beliefs for the purpose of enhancing human cognition and the usability of belief based reasoning systems.

1 Introduction

Belief theory has its origin in a model for upper and lower probabilities proposed by Dempster in [1]. Shafer later proposed a model for expressing beliefs described in his book [8]. The main idea behind belief theory is to abandon the additivity principle of probability theory, i.e. that the sum of probabilities on all pairwise exclusive possibilities must add up to one. Instead, belief theory gives observers the ability to assign so-called belief mass to any subset of the frame of discernment, i.e. non-exclusive subsets as well as the whole frame itself. The advantage of this approach is that ignorance, i.e. the lack of information, can be explicitly expressed by assigning belief mass to subsets of the frame, or to the whole frame.

While classic Dempster-Shafer belief theory represents a powerful framework for representing partial ignorance in reasoning, there is no simple connection to probability theory and statistics allowing statistical data to be interpreted as belief functions and vice versa. A more direct relationship between belief theory and probability calculus would make it possible to compute and compare expected consequences and utilities[1] of the various courses of action.

[1] See e.g. M.R.B. Clarke's response to Philippe Smets' Chapter on belief functions [9].

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 393–404, 2007.

In this paper we show how belief functions can be directly interpreted as Dirichlet probability density functions and vice versa. Models for representing uncertainty using the Dirichlet Distribution have been presented in the literature, with for example mappings to upper and lower probabilities which in some sense can represent belief functions, but none of the previously described models provide a direct mapping to belief functions on the form of bbas. This is precisely the contribution if this paper. Our method for mapping belief functions to statistical data is simple and direct. This makes it possible to perform belief based reasoning directly on statistical data, and statistical reasoning directly on belief functions. This also provides a basis for various visualizations of belief functions that facilitate human understanding of beliefs and improve the usability of belief based reasoning systems.

The remainder of this paper is organized as follows: Sec.2 gives an overview of previous approaches. Sec.3 gives an overview of the belief function theory necessary for this presentation. Then, Sec.4 presents the Dirichlet multinomial model. In Sec.5, the mapping between Dirichlet distribution and belief distribution functions is detailed, and Sec.6 describes some applications of the mapping.

2 Previous Approaches

The Imprecise Dirichlet Model (IDM) for multinomial data is described by Walley [11] as a method for determining upper and lower probabilities. The model is based on varying the base rate over all possible outcomes. The probability expectation value of an outcome resulting from assigning the total base rate (i.e. equal to one) to that outcome produces the upper probability, and the probability expectation value of an outcome resulting from assigning a zero base rate to that outcome produces the lower probability. The upper and lower probabilities are interpreted as the upper and lower bounds for the relative frequency of the outcome. While this is an interesting interpretation of the Dirichlet distribution, it can not be taken literally, as will be shown in Sec.4.

Utkin (2005) [10] defines a method for deriving beliefs and plausibilities based on the IDM, where the lower probability is interpreted as the belief and the upper probability is interpreted as the plausibility. This method can produce unreasonable results in practical applications, and Utkin provides extensions to the Imprecise Dirichlet Model to overcome some of these problems. In our view the belif and plausibility functions can not be based on the base rate uncertainty of the Dirichlet distributions. The base rates are determined by the structure of the state space when it is known, and must be estimated on a subjective basis when not known [7]. In belief theory, the state space structure is used when e.g. computing the pignistic probability expectations, but it is independent of the bba.

An indirect quantitative method for determining belief functions from statistical data is described in Shafer's book [8] (p.237). Shafer's method requires the specification of an auxiliary frame of discernment of possible probability values for the elements of the primary frame of discernment. The method can then be used to determine belief functions on the auxiliary frame of discernment based on statistical data. The awkwardness of this method makes it difficult to use in practical applications.

3 Belief Theory

In this section several concepts of the Dempster-Shafer theory of evidence [8] are recalled in order to introduce notations used throughout the article. Let $\Theta = \{\theta_i;\ i = 1, \cdots k\}$ denote a finite set of exhaustive and exclusive possible values for a state variable of interest. The frame of discernment can for example be the set of six possible outcomes of throwing a dice, so that the (unknown) outcome of a particular instance of throwing the dice becomes the state variable. A bba (basic belief assignment[2]), denoted by m, is defined as a belief distribution function from the power set 2^Θ to $[0, 1]$ satisfying:

$$m(\emptyset) = 0 \quad \text{and} \quad \sum_{x \subseteq \Theta} m(x) = 1\,. \tag{1}$$

Values of a bba are called *belief masses*. Each subset $x \subseteq \Theta$ such that $m(x) > 0$ is called a focal element of Θ.

A bba m can be equivalently represented by a non additive measure: a belief function Bel: $2^\Theta \rightarrow [0, 1]$, defined as

$$\mathrm{Bel}(x) \triangleq \sum_{\emptyset \neq y \subseteq x} m(y) \qquad \forall\, x \subseteq \Theta\,. \tag{2}$$

The quantity $\mathrm{Bel}(x)$ can be interpreted as a measure of one's total belief committed to the hypothesis that x is true. Note that functions m and Bel are in one-to-one correspondence [8] and can be seen as two facets of the same piece of information.

A few special classes of bba can be mentioned. A vacuous bba has $m(\Theta) = 1$, i.e. no belief mass committed to any proper subset of Θ. This bba expresses the total ignorance. A *Bayesian* bba is when all the focal elements are singletons, i.e. one-element subsets of Θ. If all the focal elements are nestable (i.e. linearly ordered by inclusion) then we speak about *consonant* bba. A *dogmatic* bba is defined by Smets as a bba for which $m(\Theta) = 0$. Let us note, that trivially, every Bayesian bba is dogmatic.

We will use X to denote the power set of Θ, defined by $X = 2^\Theta \backslash \Theta$, which can also be expressed as $X = \{x_i; x_i \subset \Theta\}$. Thus all proper subsets of Θ are elements of X. By considering X as a state space in itself, a general bba on Θ becomes a particular bba on X called a *Dirichlet bba*. We define $m(X) = m(\Theta)$. A belief mass on a proper subsets of Θ then becomes a belief mass on an element of X. Dirichlet bba's on X are characterised by having mutually disjoint focal elements, except the whole state space X itself. This is defined as follows.

Definition 1 (Dirichlet bba). *Let X be a state space. A bba where the only focal elements are X and/or singletons of X, is called a Dirichlet belief mass distribution function, or Dirichlet bba for short.*

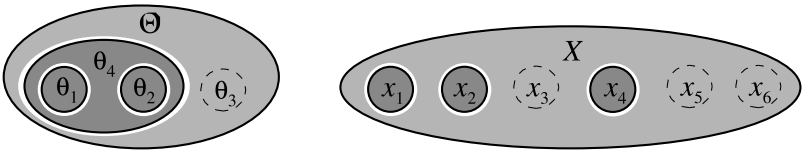
Fig.1.b below illustrates a Dirichlet bba on X, where the shaded circles around singletons and the shaded ellipse around X represent belief masses on those subsets. The focal elements in this example are X, x_1, x_2 and x_4.

[2] Called *basic probability assignment* in [8], and *Belief Mass Assignment* (BMA) in [3,4].

The number of elements in X is $|X| = 2^{|\Theta|} - 2$ when excluding $\emptyset$. For example, Fig.1.b illustrates X as having cardinality 6, meaning that it is the power set of a ternary frame of discernment. The subsets of Θ and the elements of X carry the same belief masses, so is natural to make the correspondence as simple as possible. The following example defines a possible correspondence between subsets of Θ and elements of X.

$$\begin{array}{lll} x_1 = \theta_1 & x_4 = \theta_1 \cup \theta_2 & X = \Theta \\ x_2 = \theta_2 & x_5 = \theta_1 \cup \theta_3 & \\ x_3 = \theta_3 & x_6 = \theta_2 \cup \theta_3 & \end{array} \tag{3}$$

Under this correspondence between the belief masses on X and on Θ, the focal elements of Θ are θ_1, θ_2, θ_4 and Θ as shown in Fig.1.a



(a) Focal elements of Θ (b) Focal elements of X

Fig. 1. Correspondence between belief masses on state space Θ and its power set X

The number of focal elements of a Dirichlet bba on X can be at most $|X|+1$, which happens when every element as well as X is a focal element.

The name "Dirichlet" bba is used because bba's of this type are equivalent to Dirichlet probability density functions under a specific mapping. A bijective mapping between Dirichlet bba's and Dirichlet probability density functions is defined in [5], and is also described in Section 5 below.

4 The Dirichlet Multinomial Model

The cumulative rule of combination, to be described in detail in the following sections, is firmly rooted in the classical Bayesian inference theory, and is equivalent to the combination of multinomial observations. For self-containment, we briefly outline the Dirichlet multinomial model below, and refer to [2] for more details.

4.1 The Dirichlet Distribution

We are interested in knowing the probability distribution over the disjoint elements of a state space. In case of a binary state space, it is determined by the Beta distribution. In the general multinomial case it is determined by the Dirichlet distribution, which describes the probability distribution over a k-component random variable $p(x_i),\ i = 1 \ldots k$ with sample space $[0,1]^k$, subject to the simple additivity requirement

$$\sum_{i=1}^{k} p(x_i) = 1 \ . \tag{4}$$

Because of this additivity requirement, the Dirichlet distribution has only $k-1$ degrees of freedom. This means that knowing $k-1$ probability variables and their density uniquely determines the last probability variable and its density.

The Dirichlet distribution captures a sequence of observations of the k possible outcomes with k positive real parameters $\alpha(x_i),\ i = 1 \ldots k$, each corresponding to one of the possible outcomes.

In order to have a compact notation we define a vector $\vec{p} = \{p(x_i) \mid 1 \leq i \leq k\}$ to denote the k-component random probability variable, and a vector $\vec{\alpha} = \{\alpha_i \mid 1 \leq i \leq k\}$ to denote the k-component random evidence variable $[\alpha(x_i)]_{i=1}^{k}$.

The $\vec{\alpha}$ vector represents the *a priori* as well as the observation evidence. The weight of the *a priori* evidence can be expressed as a constant C, and this weight is distributed over all the possible outcomes as a function of the base rate.

The elements in a state space of cardinality k can have a base rate different from the default value $a = 1/k$. It is thereby possible to define a base rate as a vector $\vec{a}$ with arbitrary distribution over the k mutually disjoint elements x_i with $i = 1 \ldots k$, as long as the simple additivity requirement expressed as $\sum_{i=1}^{k} a(x_i) = 1$ is satisfied. The total evidence $\alpha(x_i)$ for each element x_i can then be expressed as:

$$\alpha(x_i) = r(x_i) + C\, a(x_i)\,, \quad \text{where the constant } C \text{ is the } \textit{a priori} \text{ weight.} \tag{5}$$

The selection of the *a priori* weight C will be discussed below. The Dirichlet distribution over a set of k possible states x_i can thus be represented as a function of the base rate vector $\vec{a}$, the *a priori* weight C and the observation evidence $\vec{r}$.

Definition 2. Dirichlet Distribution

Let Θ be a state space consisting of k mutually disjoint elements. Let $\vec{r}$ represent the evidence vector over the elements of Θ and let $\vec{a}$ represent the base rate vector over the same elements. Then the multinomial Dirichlet density function over Θ can be expressed as:

$$f(\vec{p} \mid \vec{r}, \vec{a}) = \frac{\Gamma\left(\sum_{i=1}^{k} (r(x_i) + Ca(x_i))\right)}{\prod_{i=1}^{k} \Gamma\left(r(x_i) + Ca(x_i)\right)} \prod_{i=1}^{k} p(x_i)^{(r(x_i)+Ca(x_i)-1)} \tag{6}$$

where $r(x_1), \ldots r(x_k) \geq 0, \quad a(x_1), \ldots a(x_k) \in [0,1], \quad \sum_{i=1}^{k} a(x_i) = 1, \quad .$

The notation of Eq.(6) is useful, because it allows the determination of the probability distribution over state spaces where each element can have an arbitrary base rate. Given the Dirichlet distribution of Eq.(6), the probability expectation of any of the k random probability variables can now be written as:

$$\mathrm{E}(p(x_i) \mid \vec{r}, \vec{a}) = \frac{r(x_i) + Ca(x_i)}{C + \sum_{i=1}^{k} r(x_i)}\ . \tag{7}$$

It is normally required that the *a priori* distribution in case of a binary state space $X = \{x, \overline{x}\}$ is uniform, which means that $\alpha(x) = \alpha(\overline{x}) = 1$. Because of the additivity

of $\vec{a}$, then necessarily the *a priori* weight $C = 2$. Should one assume an *a priori* uniform distribution over state spaces other than binary, the constant, and also the common value would be different. The *a priori* weight C will always be equal to the cardinality of the state space over which a uniform distribution is assumed.

Selecting $C > 2$ will result in new observations having relatively less influence over the Dirichlet distribution. This could be meaningful e.g. as a representation of specific *a priori* information provided by a domain expert. It can be noted that it would be unnatural to require a uniform distribution over arbitrary large state spaces because it would make the sensitivity to new evidence arbitrarily small.

For example, requiring a uniform *a priori* distribution over a state space of cardinality 100, would force the constant C to be $C = 100$. In case an event of interest has been observed 100 times, and no other event has been observed, the derived probability expectation of the event of interest will still only be about $\frac{1}{2}$, which would seem totally counterintuitive. In contrast, when a uniform distribution is assumed in the binary case, and the same evidence is analysed, the derived probability expectation of the event of interest would be close to 1, as intuition would dictate. A good discussion of the choice of C can be found in [11].

It is here timely to revisit the Imprecise Dirichlet Model (IDM) described by Walley [11]. According to this model, the upper and lower probability values for an outcome x_i are defined as:

$$\text{IDM Upper probability:} \quad \overline{P}(x_i) = \frac{r(x_i) + C}{C + \sum_{i=1}^{k} r(x_i)} \tag{8}$$

$$\text{IDM Lower probability:} \quad \underline{P}(x_i) = \frac{r(x_i)}{C + \sum_{i=1}^{k} r(x_i)} \tag{9}$$

It can easily be shown that these values can not be literally interpreted as upper and lower bounds for for the relative frequency. For example, assume an urn containing 9 red balls and 1 black ball, meaning that the relative frequencies of red and black balls are $p(\text{red}) = 0.9$ and $p(\text{black}) = 0.1$ The *a priori* weight is set to $C = 2$. Assume further that an observer picks one ball which turns out to be black. According to Eq.(9) the lower probability is then $\underline{P}(\text{black}) = \frac{1}{3}$. It would be incorrect to literally interpret this value as the lower bound for the relative frequency because it obviously is greater than the actual relative frequency of black balls. In other words, if $\underline{P}(\text{black}) > p(\text{black})$ then $\underline{P}(\text{black})$ can impossibly be the lower bound. This case shows that the upper and lower probabilities defined by the IDM should be interpreted as an expectation value range, because that would make it consistent with the fact that actual relative frequencies can be outside the range.

4.2 Visualizing Dirichlet Distributions

Visualising Dirichlet distributions is challenging because it is a density function over $k - 1$ dimensions, where k is the state space cardinality. For this reason, Dirichlet distributions over ternary state spaces are the largest that can be practically visualised.

With $k = 3$, the probability distribution has 2 degrees of freedom, and the equation $p(x_1) + p(x_2) + p(x_3) = 1$ defines a triangular plane as illustrated in Fig.2.

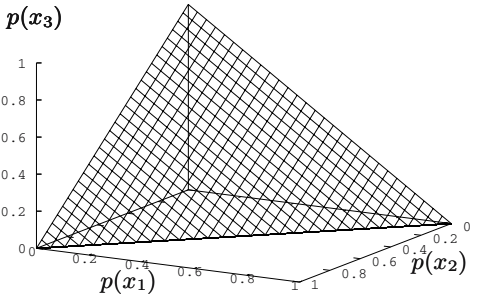


Fig. 2. Triangular plane

In order to visualise probability density over the triangular plane, it is convenient to lay the triangular plane horizontally in the X-Y plane, and visualise the density dimension along the Z-axis.

Let us consider the example of an urn containing balls of the three different types: the balls can be marked with x_1, x_2 or x_3 (i.e. $k = 3$). Let us first assume that no other information than the cardinality is available, meaning that the default base rate is $a = 1/3$, and that $r(x_1) = r(x_2) = r(x_3) = 0$. Then Eq.(7) dictates that the expected *a priori* probability of picking a ball of any specific colour is the default base rate probability, which is $\frac{1}{3}$. The *a priori* Dirichlet density function is illustrated on the left side of Fig.3.

Let us now assume that an observer has picked (with return) 6 balls of type x_1, 1 ball of type x_2 and 1 ball of type x_3, i.e. $r(x_1) = 6,\ r(x_2) = 1,\ r(x_3) = 1$, then the *a posteriori* expected probability of picking a ball of type x_1 can be computed as $\mathrm{E}(p(x_1)) = \frac{2}{3}$. The *a posteriori* Dirichlet density function is illustrated on the right side in Fig.3.

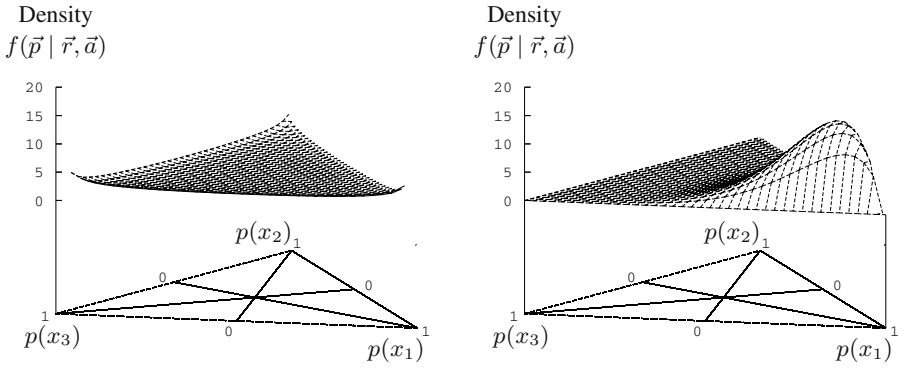
4.3 Coarsening Example: From Ternary to Binary

We reuse the example of Sec.4.2 with the urn containing red, black and yellow balls, but this time we create a binary partition of $x_1 = \{\text{red}\}$ and $x_2 = \{\text{black, yellow}\}$. The base rate of picking a red ball is set to the relative atomicity of red balls, expressed as $a(x_1) = \frac{1}{3}$.

Let us again assume that an observer has picked (with return) 6 red balls, and 2 "black or yellow" balls, i.e. $r(x_1) = 6,\ r(x_2) = 2$.

Since the state space has been reduced to binary, the Dirichlet distribution is reduced to a Beta distribution which is simple to visualise. The *a priori* and *a posteriori* density functions are illustrated in Fig.4.

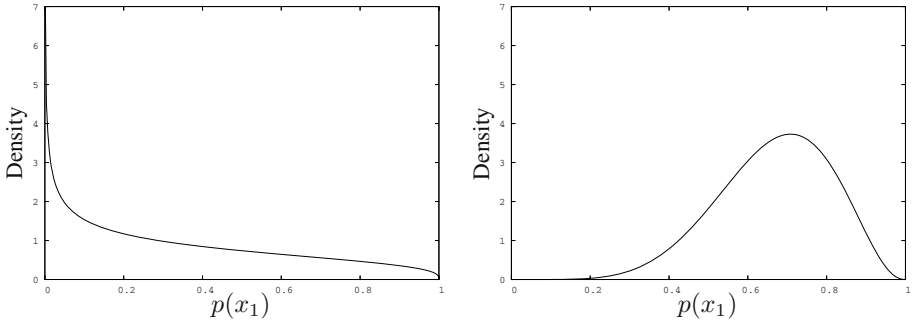
The *a posteriori* expected probability of picking a red ball can be computed with Eq.(7) as $\mathrm{E}(p(x_1)) = \frac{2}{3}$, which is the same as before the coarsening, as described in Sec.4.2. This shows that the coarsening does not influence the probability expectation value of specific events.



(a) *A priori* Dirichlet distribution

(b) Dirichlet distrib. after 6 red (x_1), 1 black (x_2) and 1 yellow (x_3) balls

Fig. 3. Visualising *a priori* and *a posteriori* Dirichlet distributions



(a) *A priori* Beta distribution, $a(x_1) = \frac{1}{3}$

(b) After 6 red and 2 yellow or black balls

Fig. 4. Visualising prior and posterior Beta distributions

5 Mapping Between Dirichlet Distribution and Belief Distribution Functions

In this section we will define a bijective mapping between Dirichlet probability distributions described in Sec.4, and Dirichlet bba's described in Sec.3.

Let $X = \{x_i;\ i = 1, \cdots k\}$ be a state space where each singleton represents a possible outcome of a state variable. It is assumed that X is the power set of a frame of discernment Θ. Let m be a general bba on Θ and therefore a Dirichlet bba on X, and let $f(\vec{p} \mid \vec{r}, \vec{a})$ be a Dirichlet probability distribution function over X.

For the bijective mapping between m and $f(\vec{p} \mid \vec{r}, \vec{a})$, we require equality between the pignistic probability values $\wp(x_i)$ derived from m, and the probability expectation values $\mathrm{E}(p(x_i))$ of $f(\vec{p} \mid \vec{r}, \vec{a})$. For all $x_i \in X$, this constraint is expressed as:

$$\wp(x_i) = \mathrm{E}(p(x_i) \mid \vec{r}, \vec{a}) \tag{10}$$

$$\Updownarrow$$

$$m(x_i) + a(x_i)m(X) = \frac{r(x_i) + a(x_i)C}{C + \sum_{t=1}^{k} r(x_t)} \tag{11}$$

We also require that $m(x_i)$ be an increasing function of $r(x_i)$, and that $m(X)$ be a decreasing function of $\sum_{t=1}^{k} r(x_t)$. In other words, the more evidence in favour of a particular outcome, the greater the belief mass on that outcome. Furthermore, the less evidence available in general, the more vacuous the bba (i.e. the greater $m(X)$). These intuitive requirements together with Eq.(11) imply the bijective mapping defined by Eq.(12).

For $m(X) \neq 0$:

$$\begin{cases} m(x_i) = \frac{r(x_i)}{C + \sum_{t=1}^{k} r(x_t)} \\ \\ m(X) = \frac{C}{C + \sum_{t=1}^{k} r(x_t)} \end{cases} \Leftrightarrow \begin{cases} r(x_i) = \frac{C m(x_i)}{m(X)} \\ \\ 1 = m(X) + \sum_{i=1}^{k} m(x_i) \end{cases} \tag{12}$$

Next, we consider the case of zero uncertainty. In case $m(X) \longrightarrow 0$, then necessarily $\sum_{i=1}^{k} m(x_i) \longrightarrow 1$, and $\sum_{i=1}^{k} r(x_i) \longrightarrow \infty$, meaning that at least some, but not necessarily all, of the evidence parameters $r(x_i)$ are infinite.

We define $\eta(x_i)$ as the the relative degree of infinity between the corresponding infinite evidence parameters $r(x_i)$ such that $\sum_{i=1}^{k} \eta(x_i) = 1$. When infinite evidence parameters exist, any finite evidence parameter $r(x_i)$ can be assumed to be zero in any practical situation because it will have $\eta(x_i) = 0$, i.e. it will carry zero weight relative to the infinite evidence parameters. This leads to the bijective mapping defined by Eq.(13).

For $m(X) = 0$:

$$\begin{cases} m(x_i) = \eta(x_i) \\ \\ m(X) = 0 \end{cases} \Leftrightarrow \begin{cases} r(x_i) = \eta(x_i) \sum_{t=1}^{k} r(x_t) = \eta(x_i)\infty \\ \\ 1 = \sum_{t=1}^{k} m(x_t) \end{cases} \tag{13}$$

In case $\eta(x_i) = 1$ for a particular evidence parameter $r(x_i)$, then $r(x_j) = \infty$ and all the other evidence parameters are finite. In case $\eta(x_j) = 1/l$ for all $j = 1 \ldots l$, then all the evidence parameters are all equally infinite.

6 Applications of the bba-Dirichlet Correspondence

Having established the mapping between Dirichlet distributions and belief mass distributions in the form of bbas, it is possible to investigate how tools from traditional probability theory can be used in belief theory and vice versa.

Bayesian updating is for example performed by simple addition of observation variables. Let $\vec{r}_A$ and $\vec{r}_B$ be two sets of observations of the same set of outcomes. The Dirichlet distribution of the combined observation is obtained by simple vector addition of $\vec{r}_A$ and $\vec{r}_B$. Mapping this vector sum to the bba space results in an operator called the cumulative fusion rule for belief [6] which represents a generalization of the consensus operator used in subjective logic [4].

Similarly, the average of statistical observations can be computed by taking the average of two sets of observations represented as vectors. Mapping the average vector to the bba space results in an operator called the averaging fusion rule for beliefs [6].

Any operator from belief theory can be translated and be applied to Dirichlet distributions, such as e.g. Dempster's orthogonal rule. Interestingly this rule becomes very different from traditional Bayesian updating. For a binary state space $X = \{x, \overline{x}\}$, Dempster's orthogonal rule can be expressed as

$$m_A \odot m_B : \begin{cases} m(x) &= \frac{m_A(x)m_B(x)+m_A(x)m_B(X)+m_A(X)m_B(x)}{1-m_A(x)m_B(\overline{x})-m_A(\overline{x})m_B(x)} \\ m(\overline{x}) &= \frac{m_A(\overline{x})m_B(\overline{x})+m_A(\overline{x})m_B(X)+m_A(X)m_B(\overline{x})}{1-m_A(x)m_B(\overline{x})-m_A(\overline{x})m_B(x)} \\ m(X) &= \frac{m_A(X)m_B(X)}{1-m_A(x)m_B(\overline{x})-m_A(\overline{x})m_B(x)} \end{cases} \tag{14}$$

The Beta distribution represents the binary case of the Dirichlet distribution. Let r represent the observation parameter of x and let s represent the observation parameter of $\overline{x}$. Bayesian updating dictates that $(r_A, s_A) + (r_B, s_B) = (r_A + r_B,\ \ s_A + s_B)$ whereas Dempster's orthogonal rule is expressed as:

$$(r_A, s_A) \odot (r_B, s_B) : \begin{cases} r = \frac{r_A r_B + 2(r_A + r_B)}{(r+s+2)^2 - r_A s_B - s_A r_B} \\ s = \frac{s_A s_B + 2(s_A + s_B)}{(r+s+2)^2 - r_A s_B - s_A r_B} \end{cases} \tag{15}$$

Combining statistical observation evidence according to the binomial Dempster's orthogonal rule of Eq.(15) is certainly new in the field of Bayesian probability theory. Generalisation to a multinomial expression is straightforward.

Belief functions on binary state spaces can be expressed as opinions in subjective logic, and visualised with the opinion triangle of subjective logic. Opinions correspond to Beta distributions which are also convenient to visualise. A simple online demonstration shows the correspondence between opinions and Beta density functions.This is shown in Fig.5, which is a screen capture of an online demonstration[3].

The example visualises opinions about three different statements x, y and z. Each belief is visualised in different ways, i.e. in the form of 1) points in an opinion triangle, 2) beta density functions, 3) coloured/shaded bars, and 4) fuzzy verbal categories.

The interpretation of the opinion triangle and the beta PDF need no further explanation, as they have been described in the previous sections. Suffice to mention that the leftmost PDF refers to x, the middle PDF refers to y and the rightmost PDF refers to z.

[3] http://www.fit.qut.edu.au/ josang/sl/demo/BV.html

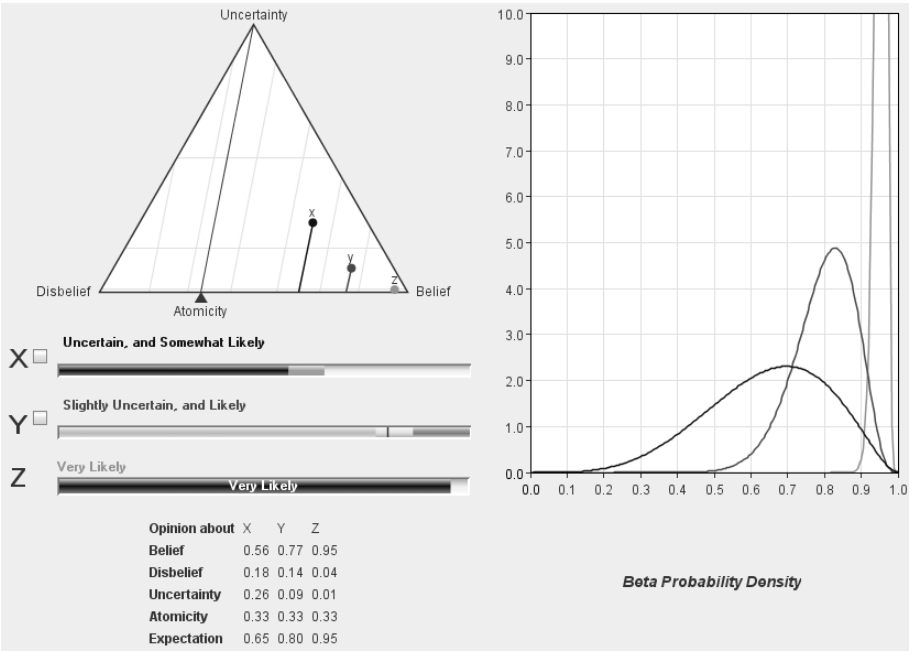


Fig. 5. Example visualisations of binomial opinions

The horizontal shaded bars are actually coloured in the online demonstration, which makes them easier to interpret. The first horizontal bar, representing the belief in x, consists of a dark shaded area representing b_x, and a light shaded area representing $a_x u_x$ (i.e. the amount of uncertainty that contributes to $\mathrm{E}(x)$, so that the total length of the dark and light shaded areas together represent $\mathrm{E}(x)$.

The second horizontal bar, representing the belief in y, consists of a green (leftmost) area representing b_y, an amber (middle) area representing u_y, and a red (rightmost) area representing d_y, as well as a black vertical line within the amber area indicating $\mathrm{E}(y)$. The second horizontal bar thus uses the traffic light metaphor, where green indicates "go", red indicates "stop" and amber indicates "caution".

The third horizontal bar, representing the belief in z, simply has a single dark shaded area representing $\mathrm{E}(z)$.

7 Conclusion

The mapping between beliefs and probability distribution functions puts belief theory and statistical theory firmly together. This is important in order to make belief theory more practical and easier to interpret, and to make belief theory more acceptable in the main stream statistics and probability communities.

References

1. Dempster, A.P.: Upper and lower probabilities induced by a multiple valued mapping. Annals of Mathematical Statistics 38, 325–339 (1967)
2. Gelman, A., et al.: Bayesian Data Analysis, 2nd edn. Chapman and Hall/CRC, Florida (2004)
3. Jøsang, A.: A Logic for Uncertain Probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9(3), 279–311 (2001)
4. Jøsang, A.: The Consensus Operator for Combining Beliefs. Artificial Intelligence Journal 142(1-2), 157–170 (2002)
5. Jøsang, A.: Probabilistic Logic Under Uncertainty. In: In the Proceedings of Computing: The Australian Theory Symposium (CATS2007), CRPIT , Ballarat, Australia, vol. 65 (January 2007)
6. Jøsang, A., Marsh, S.: The Cumulative Rule of Belief Fusion. In: The Proceedings of the International Conference on Information Fusion, Quebec, Canada. July 2007 (to appear)
7. Pope, S., Jøsang, A.: Analsysis of Competing Hypotheses using Subjective Logic. In: Proceedings of the 10th International Command and Control Research and Technology Symposium (ICCRTS). United States Department of Defense Command and Control Research Program (DoDCCRP) (2005)
8. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
9. Smets, Ph.: Belief Functions. In: Smets, Ph., et al. (eds.) Non-Standard Logics for Automated Reasoning, pp. 253–286. Academic Press, London (1988)
10. Utkin, L.V.: Extensions of belief functions and possibility distributions by using the imprecise Dirichlet model. Fuzzy Sets and Systems 154, 413–431 (2005)
11. Walley, P.: Inferences from Multinomial Data: Learning about a Bag of Marbles. Journal of the Royal Statistical Society 58(1), 3–57 (1996)

Forward-Backward-Viterbi Procedures in the Transferable Belief Model for State Sequence Analysis Using Belief Functions

Emmanuel Ramasso, Michèle Rombaut, and Denis Pellerin

GIPSA-lab, Images and Signal Department,
46 av. Félix Viallet, 38031 Grenoble, France
e_ramasso@yahoo.fr,
{rombaut,pellerin}@gipsa-lab.inpg.fr

Abstract. The Transferable Belief Model (TBM) relies on belief functions and enables one to represent and combine a variety of knowledge from certain up to ignorance as well as conflict inherent to imperfect data. A lot of applications have used this flexible framework however, in the context of temporal data analysis of belief functions, a few work have been proposed. Temporal aspect of data is essential for many applications such as surveillance (monitoring) and Human-Computer Interfaces. We propose algorithms based on the mechanisms of Hidden Markov Models usually used for state sequence analysis in probability theory. The proposed algorithms are the "credal forward", "credal backward" and "credal Viterbi" procedures which allow to filter temporal belief functions and to assess state sequences in the TBM framework. Illustration of performance is provided on a human motion analysis problem.

1 Introduction

Analysis of state sequence is important in many fields such as Signal Processing and Computer Vision [1,2,3]. State sequence analysis is generally performed in Bayesian framework using Hidden Markov models (HMM) [1,2] where probabilities are used to handle uncertainty on states. In HMM, one can only observe some features related to states but not the states directly in part because of noise. Given a sequence of noise observations, HMM machinery is able to retrieve the best sequence of states using a Viterbi decoding [1] relying on a forward propagation scheme. The latter is used for state filtering (online) whereas smoothing (offline) is performed using a backward procedure. Particular combinations of forward–backward procedures allow to estimate HMM parameters such as the state transition matrix [1]. Usual HMM can only be applied on probabilities.

Transferable Belief Model (TBM) [4] can model more complex cases of uncertainty than probabilities. It is a general model used to represent and combine a variety of knowledge. In particular, doubt and conflict are explicitly emphasized. Doubt smartly represents ignorance (useful to initialize HMM and to represent state transition) and conflict emphasizes the contradiction within a fusion process (can be exploited for state sequence analysis [5]).

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 405–417, 2007.

The TBM has seldom been used for temporal data and state sequence analysis in a noise context. In this paper a method is proposed for this purpose and that we call credal HMM (CrHMM). The CrHMM combines the genericity of TBM and mechanisms of HMM. The idea to generalize HMM to evidence theory was initiated by Pieczynski et al. (see [6] for a recent work) but the generalization is based on Dempster's rule of combination (with normalization) and assumes that either the prior or (exclusively) the observation is evidential (and generally a *simple* BBA obtained by discounting). Therefore, the combination yields a Bayesian belief function. The formalism proposed in this paper handles general belief functions (as understood in TBM), is strongly based on conflict and relies on Smets' work concerning Evidential Network [7,8,9]. Moreover, a first *credal Viterbi* algorithm is proposed.

Credal forward-backward algorithms are expressed by commonalities in order to mimick their counterpart in HMM. Then, particular combinations of credal forward-backward algorithms are presented for CrHMM learning. Lastly, a credal Viterbi decoding algorithm is proposed to retrieve the best sequence of states when knowledge on observations and priors is modelled by belief functions. Both the credal forward and the credal Viterbi decoding generate one criterion (based on conflict information) used for inference in the context of competing CrHMMs. Illustrations of algorithms capabilities concern human motion analysis in videos.

Section 2 presents HMM machinery. Section 3 describes some TBM's tools used in this paper. Section 4 is devoted to the credal version of the forward algorithm. Section 5 focuses on the extension of the backward algorithm and on some variables useful for learning credal HMM. Section 6 presents the credal Viterbi decoding. Lastly, illustrations are provided.

2 Hidden Markov Models: Basics

In this section, main elements of HMM are recalled. For the remainder of this paper, we assume the reader is familiar with basics in HMM. The reader can read the well known tutorial of Rabiner [1] for details. We will refer to it several times in this paper. All conditions of independance are assumed to be satisfied in both the probabilistic [1] and credal cases [7,10]. Fig. 1 depicts forward and

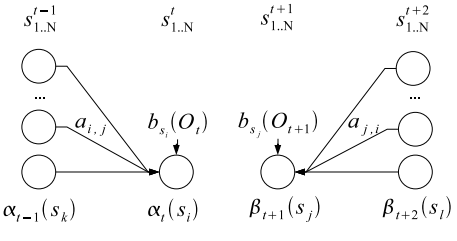


Fig. 1. Computation (from [1]) of forward (α_t) and backward (β_t) variables using past and observations likelihood (b_{s_i}). Circles for states and arrows for propagation direction.

backward processes explained hereafter. This figure will be widely used for the generalization in TBM framework.

An HMM is a stochastic state machine for the recognition of state sequence from observations. A sequence is supposed to be composed of N states s_i^t at time $t \in \{1 \dots T\}$ and at any time $s_i^t \in \Omega_t$ where Ω_t is called frame of discernment (FoD) defined by $\Omega_t = \{s_1^t, s_2^t, \dots s_N^t\}$. The time t will be explicitly denoted as superscript of states when required, for instance s_i^t is the state s_i at t. At a given time t, states are said *hidden* and only observations denoted $\mathbf{O}_t \in \Re^F$ are effectively measured, one at each time t. Typically, observations $\mathbf{O}_t$ represent a set of F features values. Models of observations are used to infer the likelihood l of states given observations, i.e. [1] $b_{s_i}(\mathbf{O}_t) = l(s_i^t|\mathbf{O}_t) = P(\mathbf{O}_t|s_i^t)$. These likelihoods are generally provided by a mixture of Gaussians (MoG) for each state [1]. Then, inference on Ω_t is performed by Bayes' theorem providing the posterior probability of state s_i given observations and priors. Sequence analysis by an HMM requires a *transition matrix* $A = [a_{ij}] = P(s_j^t|s_i^{t-1})$ (Markovian assumption), a *prior* π (π_i for state s_i) and *observation models* (MoG). These three elements represent a particular HMM λ. In case distinct sequences are to be recognized, one particular HMM λ is necessary for each sequence. An HMM is used for several tasks [2]: online filtering (forward variable α), offline smoothing (γ-variable obtained from backward β and α), learning (ξ-variable obtained from β, α and γ) and evaluation (Viterbi decoding δ) of states sequences.

3 TBM Background

In this section are recalled some basics of TBM used to derive the CrHMM.

3.1 Basic Concepts

An agent's belief is expressed on a finite FoD Ω_t and is represented by a basic belief assignment (BBA) m^{Ω_t} from 2^{Ω_t} to $[0, 1]$ with respect to $\sum_{S_i \subseteq \Omega_t} m^{\Omega_t}(S_i) = 1$. For the sake of simplicity, braces around sets will be forgotten, $\{s_1^t, s_3^t\} = \{s_1^t\} \cup \{s_3^t\} \equiv s_1^t \cup s_3^t$. Belief functions are non-additive measures and this allows to explicitly model doubt between states. This is a fundamental difference with probability theory. In the sequel, a singleton is denoted with small cap (e.g. $s_i^t \in \Omega_t$) whereas big cap is used for union of singletons (e.g. $S_i^t \subseteq \Omega_t$).

A BBA m^{Ω_t} can be converted into plausibility pl^{Ω_t} and commonality q^{Ω_t} functions. These functions represent belief in different ways, emphasize some particularities and allow to decrease some computational aspects. We denote $f^{\Omega_t} \in \{q^{\Omega_t}, pl^{\Omega_t}, m^{\Omega_t}\}$ these functions which are in one-to-one correspondance. The relations that we will use are defined $\forall S_i, S_k \subseteq \Omega_t$ by:

$$pl^{\Omega_t}(S_i) = \sum_{S_i \cap S_k \neq \emptyset} m^{\Omega_t}(S_k) \tag{1}$$

$$m^{\Omega_t}(S_i) = \sum_{S_k \supseteq S_i} (-1)^{|S_k| - |S_i|} q^{\Omega_t}(S_k) \tag{2}$$

$$m^{\Omega_t}(S_i) = \sum_{S_k \supseteq S_i} (-1)^{|S_k|-|S_i|} q^{\Omega_t}(S_k) \tag{3}$$

$$q^{\Omega_t}(S_i) = \sum_{S_k \subseteq S_i} (-1)^{|S_k|+1} pl^{\Omega_{t+1}}(S_k) \tag{4}$$

3.2 Combination, Extension and Marginalization

Given two distinct [10] BBAs, $m_1^{\Omega_t}$ and $m_2^{\Omega_t}$ defined on the same FoD Ω_t, the *conjunctive rule of combination* (CRC) is defined $\forall S_i, S_k, S_l \subseteq \Omega_t$ by:

$$(m_1^{\Omega_t} \textcircled{\cap} m_2^{\Omega_t})(S_l) = \sum_{S_i \cap S_k = S_l} m_1^{\Omega_t}(S_i) \cdot m_2^{\Omega_t}(S_k) \tag{5}$$

and equivalently $(q_1^{\Omega_t} \textcircled{\cap} q_2^{\Omega_t})(S_l) = q_1^{\Omega_t}(S_l) \cdot q_2^{\Omega_t}(S_l)$ (commonalities simplifies computation). The *disjunctive rule of combination* (DRC) is [11,7]:

$$(m_1^{\Omega_t} \textcircled{\cup} m_2^{\Omega_t})(S_l) = \sum_{S_i \cup S_k = S_l} m_1^{\Omega_t}(S_i) \cdot m_2^{\Omega_t}(S_k) \tag{6}$$

In case FoDs are different, e.g. one wants to combine m^{Ω_t} and $m^{\Omega_{t+1}}$, then BBAs must be *extended* on the common FoD $\Omega_t \times \Omega_{t+1}$ before combination using the so-called vacuous extension [7] denoted "$\uparrow$". E.g. m^{Ω_t} is redefined as:

$$m^{\Omega_t \uparrow \Omega_t \times \Omega_{t+1}}(S) = \begin{cases} m^{\Omega_t}(S_i) \text{ if } S = S_i \times \Omega_{t+1} \\ \qquad\qquad \text{and } S_i \subseteq \Omega_t \\ 0 \qquad\quad \text{otherwise} \end{cases} \tag{7}$$

After combination by the CRC or by the DRC, the result $m^{\Omega_t \times \Omega_{t+1}}$ can be marginalized onto Ω_{t+1} or Ω_t. Assuming the marginal is computed on Ω_{t+1}, the marginalization operator, denoted "$\downarrow$", is defined $\forall S_j \subseteq \Omega_{t+1}$ by:

$$m^{\Omega_t \times \Omega_{t+1} \downarrow \Omega_{t+1}}(S_j) = \sum_{\substack{S \subseteq \Omega_t \times \Omega_{t+1} \\ S^{\downarrow \Omega_{t+1}} = S_j}} m^{\Omega_t \times \Omega_{t+1}}(S) \tag{8}$$

with "$S^{\downarrow \Omega_{t+1}} = S_j$" means S_j is the projection of S on Ω_{t+1}.

3.3 Conditioning and Ballooning Extension

In a state sequence, conditional beliefs describes how true can be the states in $S_j \subseteq \Omega_{t+1}$ given the previous states in $S_i \subseteq \Omega_t$. Given two BBAs: $m_i^{\Omega_t}$ defined by $m_i^{\Omega_t}(S_i) = 1$ and $m^{\Omega_t \times \Omega_{t+1}}$, the conditional belief $m^{\Omega_{t+1}}[S_i^t]$ is defined by:

$$m^{\Omega_{t+1}}[S_i^t] = \left(m_i^{\Omega_t \uparrow \Omega_t \times \Omega_{t+1}} \textcircled{\cap} m^{\Omega_t \times \Omega_{t+1}} \right)^{\downarrow \Omega_{t+1}} \tag{9}$$

Conversely, if a conditional BBA $m^{\Omega_{t+1}}[s_i^t]$ is provided $\forall s_i \in \Omega_t$, it is possible to cancel the conditioning revision using the *ballooning extension* [7]. Let us denote

$v_j = ((s_i \times \Omega_{t+1}) \cap S)^{\downarrow \Omega_{t+1}})$ (where $s_i \times \Omega_{t+1}$ is called cylindrical extension of state (singleton) $s_i \in \Omega_t$ on $\Omega_t \times \Omega_{t+1}$), then Smets proves [7]:

$$q^{\Omega_t \times \Omega_{t+1}}(S) = \prod_{s_i \in \Omega_t} q^{\Omega_{t+1}}[s_i](v_j) \tag{10}$$

Ballooning extension and vacuous extension [7] are used in this paper in order to compute a belief on a joint space $\Omega_t \times \Omega_{t+1}$.

3.4 The Generalized Bayesian Theorem

Bayes' theorem was extended in TBM framework by Smets [7] and is called the Generalized Bayesian Theorem (GBT). The GBT alleviates the problem of priors since belief functions allow to represent total ignorance.

Definition. Given a vacuous prior ($m^{\Omega_t}(\Omega) = 1$) and the set of conditional beliefs $f^{\Omega_{t+1}}[s_i]$, the posterior $f^{\Omega_{t+1}}(S_j)$, $\forall S_j \in \Omega_{t+1}$ is:

$$f^{\Omega_{t+1}}(S_j) = \sum_{S_i \subseteq \Omega_t} f^{\Omega_{t+1}}[S_i](S_j) \cdot m^{\Omega_t}(S_i) \tag{11}$$

Since the conditional belief is initially given conditionally to each singleton state $s_i \in \Omega_t$, the belief defined conditionally to a subset S_i is obtained by the DRC (Eq. (6)) assuming distinctness [7]:

$$f^{\Omega_{t+1}}[S_i] = \bigcirc\!\!\!\!\!\!\bigcup_{s_i \in S_i} f^{\Omega_{t+1}}[s_i], \quad \forall S_i \subseteq \Omega_t \tag{12}$$

Eq. (11)-(12) are the core of Evidential Networks [7,8,10].

Motivation: From Likelihood to Belief Function. Assume a set of features $\mathbf{O}_t$ taking values in $\Re^F$. As emphasized in [12], often the conditional belief over $\Re^F$ given s_i is represented by a probability function. In this case $pl^{\Re^F}[s_i](\mathbf{O}_t) = P(\mathbf{O}_t|s_i) = l(s_i|\mathbf{O}_t)$, so the vector of plausibilities equals the vector of likelihoods of s_i given $\mathbf{O}_t$ [7,12]. Given the likelihoods $l(s_i|\mathbf{O}_t)$ for each $s_i \in \Omega_t$, then for $\mathbf{O}_t \in \Re^F$ and for each $S \subseteq \Omega_t$, Smets [7] proves:

$$q_b^{\Omega_t}[\mathbf{O}_t](S) = \prod_{s_i \in S} l(s_i|\mathbf{O}_t) \tag{13}$$

where $q_b^{\Omega_t}[\mathbf{O}_t]$ is the posterior commonality conditionally to $\mathbf{O}_t$ and defined on Ω_t. It is the counterpart of the probability $b_{s_i}(\mathbf{O}_t)$ [1] but now the commonality is defined for union of states.

3.5 Decision-Making

After combination of multiple sources of belief, a resulting BBA m^{Ω_t} is obtained. Decision-making under uncertainty and imprecision based on belief functions

must be made either on the pignistic probability [7] or on plausibilities [12], both assign a value to each element (singleton) $s_i \in \Omega_t$. In the sequel, the pignistic probability is used and it is defined by:

$$\mathbf{BetP}\{m^{\Omega_t}\}(s_i) = \sum_{S_k \subseteq \Omega_t} \frac{|s_i \cap S_k|}{|S_k|} \frac{m^{\Omega_t}(S_k)}{1 - m^{\Omega_t}(\emptyset)} \tag{14}$$

where $m^{\Omega_t}(\emptyset)$ is the conflict value and $1 - m^{\Omega_t}(\emptyset)$ is a normalizing coefficient.

4 Filtering Belief Functions: Credal Forward Algorithm

In HMM, the forward algorithm allows to filter (online) probabilities that evolves along time. The forward algorithm relies on the forward variable generally denoted $\alpha(t)$ [1].

A similar *credal forward algorithm* can be obtained. Commonalities are used in order to mimick their counterpart based on probabilities. The algorithm is a simple propagation scheme that follows left part of Fig. 1. The credal forward algorithm consists of three steps:

1. **Initialization :** $\forall S_i \subseteq \Omega_1$, apply Eq. (15).
2. **Induction:** $1 \leq t \leq T-1$, $S_j \subseteq \Omega_{t+1}$, apply Eq. (16).
3. **Termination:** apply Eq. (17).

We denote $f_\alpha^{\Omega_t} \in \{q_\alpha^{\Omega_t}, pl_\alpha^{\Omega_t}, m_\alpha^{\Omega_t}\}$ the credal version of the forward variable. Subscripts α, a and b are used to mimick, respectively, the forward variable, transition probabilities and observations likelihood as defined for usual HMM [1].

4.1 Initialization

One asset of using belief functions is the possibility to explicitly model a vacuous prior for $t = 1$ (no prior):

$$q_\alpha^{\Omega_{t=1}}(S_i) = 1, \quad \forall S_i \subseteq \Omega_t \tag{15}$$

or equivalently $m_\alpha^{\Omega_{t=1}}(\Omega) = 1$. This lack of information is well handled and smartly represented using belief functions whereas probabilities require priors. One can also use more complex initialization such as consonant belief functions.

4.2 Induction

Induction can be retrieved from Smets' work on Evidential Network [7,8]. It relies on the credal forward variable. Given:

1. $m_\alpha^{\Omega_t}$ the BBA of the forward variable of states at t,
2. $q_a^{\Omega_{t+1}}[S_i]$, $\forall S_i \subseteq \Omega$ the set of conditional commonality distribution which links states and set of states from t to $t+1$ (obtained from Eq. (11) and (12)),

3. $q_b^{\Omega_{t+1}}[\mathbf{O}_t]$ the observations, obtained e.g. from a mixture of Gaussians and GBT (Eq. (13)) at $t+1$.

The credal forward variable is a commonality that combines past information, transition and current observations (left part of Fig. 1) by:

$$q_\alpha^{\Omega_{t+1}}(S_j) = \Big(\sum_{S_i \subseteq \Omega_t} m_\alpha^{\Omega_t}(S_i) \cdot q_a^{\Omega_{t+1}}[S_i](S_j) \Big) \cdot q_b^{\Omega_{t+1}}[\mathbf{O}_t](S_j) \tag{16}$$

defined $\forall S_j \subseteq \Omega_{t+1}$. This equation has a close form compared to the probability-based forward pass but this one works on belief functions and on sets of states. The first part is the application of the GBT with priors ($m_\alpha^{\Omega_t}(S_i)$) and conditional commonalities ($q_a^{\Omega_{t+1}}[S_i](S_j)$). The second part represents observations ($q_b^{\Omega_{t+1}}[\mathbf{O}_t](S_j)$) conjunctively combined using the CRC.

4.3 Termination Step

In HMM, the termination step of the forward algorithm [1] is $\sum_{s_i \in \Omega_T} \alpha_T(s_i) = P(\mathbf{O}_{1:T}|\lambda)$ and represents the state sequence probability. However, we can not apply the same termination step with the credal version because belief on focal sets of $m_\alpha^{\Omega_t}$ always sum to 1 for any t. Actually, the BBA obtained at T does not reflect the whole sequence but only the belief on states at T. Instead, we propose to exploit conflict for state sequence analysis. The proposed criteria has a similar role to the log-likelihood used for MAP classification. Given several HMM $\lambda_1 \dots \lambda_k \dots \lambda_K$, the best model λ^* explaining observations on $[1, T]$ minimizes conflict along the sequence of observations $\mathbf{O}_{1:T}$:

$$L_c^1(\lambda) = \frac{1}{T} \sum_{t=1}^{T} \log(m_\alpha^{\Omega_t}[\lambda](\emptyset)) \tag{17}$$

$$\lambda^* = \underset{k}{\operatorname{argmin}}\ L_c^1(\lambda_k) \tag{18}$$

During induction, we keep track of this conflict and then normalize the BBA. Eq. (17) is similar to the one proposed and justified by Ristic and Smets [13] where it is used as a distance measure between objets for association purpose. Using commonalities, the computation in the credal forward algorithm is close to the probability-based version [1] but the former is greedy (see [14,7] for computational solutions).

5 Smoothing and Towards Learning

State sequence can be filtered offline, this is called smoothing [2]. Smoothing is used for learning HMM parameters. Rabiner [1] presents three variables used for smoothing/learning: the backward (β) variable (as in right part of Fig. 1), the γ-variable and the ξ-variable. The first one is generally combined with the

forward variable for offline smoothing. The ξ-variable is exploited for learning the transition matrix of HMM. Learning the transition matrix includes an iterated procedure that we will not be developed for the CrHMM. We only propose the equivalent expressions of the three variables.

5.1 The Credal Backward Induction

Likewise to the credal forward variable, the credal backward variable is computed using belief propagation and following the right part of Fig. 1. The credal backward induction is defined $\forall S_i \subseteq \Omega_t$ by:

$$q_\beta^{\Omega_t}(S_i) = \sum_{S_j \subseteq \Omega_{t+1}} \Big(\big(m_\beta^{\Omega_{t+1}} \textcircled{\cap} m_b^{\Omega_{t+1}}[\mathbf{O}_t]\big)(S_j) \cdot q_a^{\Omega_t}[S_j](S_i) \Big) \tag{19}$$

where $\big(m_\beta^{\Omega_{t+1}} \textcircled{\cap} m_b^{\Omega_{t+1}}[\mathbf{O}_t]\big)(S_j)$ is the value of the BBA on set S_j resulting from the combination by the CRC of both $m_\beta^{\Omega_{t+1}}$ and $m_b^{\Omega_{t+1}}[\mathbf{O}_t]$. Since only $q_a^{\Omega_{t+1}}[S_i](S_j)$ is known (i.e. the conditional belief of proposition S_j at $t+1$ given S_i at t), it is required to derive $q_a^{\Omega_{t+1}}[S_j](S_i)$ from it. For that, we use the Generalized Likelihood Principle [7] which postulates that $pl_a^{\Omega_{t+1}}[S_i](S_j) = pl_a^{\Omega_t}[S_j](S_i)$, where $pl_a^{\Omega_{t+1}}[S_i](S_j)$ is the conditional plausibility (Eq. (1)) of being in state S_j at $t+1$ given S_i at t and which is known. From plausibility $pl_a^{\Omega_t}[S_j]$, commonality $q_a^{\Omega_t}[S_j]$ is derived by Eq. (4). The backward variable is initialized with a vacuous prior.

5.2 The Credal γ Variable

The joint observation of both the credal forward and backward variables can be obtained by the CRC:

$$q_\gamma^{\Omega_t} = q_\alpha^{\Omega_t} \textcircled{\cap} q_\beta^{\Omega_t} \tag{20}$$

obtained $\forall S_i \subseteq \Omega_t$. The credal variable $q_\gamma^{\Omega_t}$ might be useful for learning HMM parameters in the credal case. Another role of $q_\gamma^{\Omega_t}$ is to assess the best state s_t^* at t of the current sequence by $s_t^* = \text{argmax}_{\,s_i \in \Omega_t} \mathbf{BetP}\{m_\gamma^{\Omega_t}\}(s_i)$ for $1 \leq t \leq T$ and where **BetP** is defined by Eq. (14).

5.3 The Credal ξ Variable

Since the probability-based ξ variable is used to learn the transition matrix [1], it would be interesting to define it in TBM framework for further learning. In case learning might be done online or offline, we propose hereafter two versions of the credal ξ variable: one to estimate it on-the-fly and denoted $f_{\xi_{\text{on}}}^{\Omega_{t-1} \times \Omega_t}$ and one for the off-line case denoted $f_{\xi_{\text{off}}}^{\Omega_t \times \Omega_{t+1}}$, both defined on the product space of two successive time slices. The joint space allows to explicitly model the link (transition) between each couple of states likewise to the transition matrix but the credal version explicitly models doubt between states.

On-Line Estimation. In the online case, only the available information up to time t are combined. Thus it is based on the credal forward algorithm, observations (at t), priors (if available, at $t-1$) and conditional beliefs (t given $t-1$). We define it for $S_i \subseteq \Omega_{t-1}$, $S_j \subseteq \Omega_t$ and $S \subseteq \Omega_t \times \Omega_{t-1}$ as:

$$q_{\xi_{\text{on}}}^{\Omega_{t-1}\times\Omega_t}(S) = q_{\alpha}^{\Omega_{t-1}\uparrow\Omega_{t-1}\times\Omega_t}(S) \times q_{a}^{\Omega_{t-1}\times\Omega_t}(S) \times q_{b}^{\Omega_t}[\mathbf{O}_t]^{\uparrow\Omega_{t-1}\times\Omega_t}(S) \quad (21)$$

and where $q_{a}^{\Omega_{t-1}\times\Omega_t}(S)$ is computed by Eq. (10). Moreover, the marginalization of $q_{\xi_{\text{on}}}^{\Omega_{t-1}\times\Omega_t}$ onto Ω_t results in the forward variable, i.e. $q_{\alpha}^{\Omega_t} = q_{\xi_{\text{on}}}^{\Omega_{t-1}\times\Omega_t\downarrow\Omega_t}$.

Off-Line Estimation. For the offline case, both backward (up to $t+1$) and forward (up to t) propagations are combined. The link between t and $t+1$ is made by the conditonal belief. We define it $\forall S \subseteq \Omega_t \times \Omega_{t+1}$ by:

$$\begin{aligned} q_{\xi_{\text{off}}}^{\Omega_t\times\Omega_{t+1}}(S) = {} & q_{\alpha}^{\Omega_t\uparrow\Omega_t\times\Omega_{t+1}}(S) \times q_{\beta}^{\Omega_{t+1}\uparrow\Omega_t\times\Omega_{t+1}}(S) \\ & \times q_{b}^{\Omega_{t+1}}[\mathbf{O}_t]^{\uparrow\Omega_t\times\Omega_{t+1}}(S) \times q_{a}^{\Omega_t\times\Omega_{t+1}}(S) \end{aligned} \quad (22)$$

There is one commonality $q_{\xi_{\text{off}}}^{\Omega_t\times\Omega_{t+1}}$ at each time. The joint form of the conditional belief is obtained by applying Eq. (10) on $q_{a}^{\Omega_{t+1}}[S_j](S_i)$ as explained Section 5.1 for the backward variable. Note that marginalizing $q_{\xi_{\text{off}}}^{\Omega_t\times\Omega_{t+1}}$ onto Ω_{t+1} yields the credal γ variable, i.e. $q_{\gamma}^{\Omega_{t+1}} = q_{\xi_{\text{off}}}^{\Omega_t\times\Omega_{t+1}\downarrow\Omega_{t+1}}$.

6 Credal Viterbi Procedure

The goal is to determine which state $s_i \in \Omega_{t-1}$ at $t-1$ (*current candidate*) accounts for the ended-state sequence s_j at time t (*current hypothesis*).

State-of-knowledge between t and $t-1$ is given by Eq. (21). Therefore, likewise to the probabilistic case, the credal Viterbi relies on the forward pass. Since it is required to test each hypothesis $s_j \subseteq \Omega_t$, the BBA obtained from Eq. (21) and (3) is conditioned on the current hypothesis s_j. Moreover, it is necessary to determine which state s_i is the best candidate therefore, the conditioning result is marginalized onto the space Ω_{t-1} where the current candidate s_i is defined. Hence, the BBA used for decision is formally defined by:

$$m_{\text{dec}}^{\Omega_{t-1}}[s_j] = m_{\xi_{\text{on}}}^{\Omega_{t-1}\times\Omega_t}[s_j]^{\downarrow\Omega_{t-1}} \quad (23)$$

Conditioning is equivalent to a conjunctive combination with a categorical belief mass of the form $m_j^{\Omega_{t-1}\times\Omega_t}(s_j) = 1$ (Section 3.3). Therefore, a conflict may appear and quantifies the *incoherence of being in state s_j at t*. This conflict is used in the decision process and it is the core of the credal Viterbi procedure. Let us denote $\mathbf{C}_t(s_j)$ the value of the coherence at time t:

$$\mathbf{C}_t(s_j) = 1 - m_{\text{dec}}^{\Omega_{t-1}}[s_j](\emptyset) \quad (24)$$

Decision must be made on the pignistic probability distribution (Eq. (14)) defined over Ω_{t-1} from Eq. (23). This space, Ω_{t-1}, is characterized by a set of coherences $\mathbf{C}_{t-1}(s_i)$ (s_i is used because time index is $t-1$) computed at the previous time slice from Eq. (24). Therefore, coherences are taken into account in the decision process performed at time t on space Ω_{t-1} by weighting the pignistic probabilities defined over Ω_{t-1}:

$$P_t[s_j](s_i) = \mathbf{C}_{t-1}(s_i) \cdot \mathbf{BetP}\{m_{\text{dec}}^{\Omega_{t-1}}[s_j]\}(s_i) \tag{25}$$

It is required to normalize the coherences in order to obtain a probability distribution. The BBA $m_{\text{dec}}^{\Omega_{t-1}}[s_j]$ is obtained at time t from knowledge at both t and $t-1$ (it is projected onto Ω_{t-1}) whereas coherences concern time index $t-1$ (not t) obtained at the previous time. From the obtained (weighted) probability distribution (Eq. (25)), the decision (choosing the best candidate s_i) is made.

Lastly, we compute another log-contradiction criteria $L_c^2(\lambda)$ using the results of the credal Viterbi procedure (see step 4b)). Indeed, when the path is recovered, we take into account the coherence along the sequence. The basic idea is that the lower the conflict along the path, the better the sequence corresponds to the model λ. We define it in the following algorithm:

1. **Initialization :** $\forall s_i \in \Omega_1$, $\mathbf{C}_1(s_i) = 1$, $L_c^2(\lambda) = 0$ and $\psi^{\Omega_1}(s_i) = 0$
2. **Recursion:** $2 \leq t \leq T-1$, $\forall s_j \in \Omega_t$, $\forall s_i \in \Omega_{t-1}$
   a) Compute $m_{\text{dec}}^{\Omega_{t-1}}[s_j]$ with Eq. (23)
   b) $P_t[s_j](s_i) = \mathbf{C}_{t-1}(s_i) \cdot \mathbf{BetP}\{m_{\text{dec}}^{\Omega_{t-1}}[s_j]\}(s_i)$
   c) $\psi_t(s_j) = \text{argmax}_{\,s_i \in \Omega_{t-1}} [P_t[s_j](s_i)]$
   d) Compute $\mathbf{C}_t(s_j)$ with Eq. (24)
3. **Termination:** $s_T^* = \text{argmax}_{\,\forall s_j \in \Omega_T}[\text{max}_{\forall s_i \in \Omega_{T-1}} P_T[s_j](s_i)]$
4. **Path Backtracking:**
   a) $s_t^* = \psi_{t+1}(s_{t+1}^*)$
   b) $L_c^2(\lambda) \leftarrow L_c^2(\lambda) + \log(1 - \mathbf{C}_t(s_t^*))$

where "$\leftarrow$" is an assignment in order to update the second proposed log-contradiction criteria. The variable $\mathbf{C}_t$ is computed at t for all $s_j \subseteq \Omega_t$ (line d)) and is used for the next step.

Transition and Ambiguity Detection. The coherence measure is used for states transition detection. When the decision is made (step 3)), one can compel the probability (step 2b)) to be greater than a given threshold σ in order to trust the decision (e.g. $\sigma = 1/N$ with N the number of states).

7 Illustration: Human Motion Analysis

Our aim is to *illustrate* some of the proposed algorithms : smoothing, evaluation and classification *a posteriori*. For that, we use data of previous communications [15,16] that concern the recognition of three athletics jumps (activity):

long jumps, pole vaults and high jumps. For this paper, each activity is modelled by one HMM and one CrHMM. Each model of activity is made of four states (actions) with $\Omega_t = \{\text{running}, \text{jumping}, \text{falling}, \text{standing}\}$. Given a set of features extracted at each video frame, a likelihood on each action is computed using a mixture of gaussians. Likelihoods at each video frame are transformed into belief on actions using Eq. (13). HMM parameters are learned using Baum-Welch algorithm [1] on the half of the database. Learning CrHMM is still in its infancy: actually we proposed some equations in previous sections that need to be embedded in an iterated procedure such as CrEM [17] in order to compute it. The credal transition matrix is thus computed from the probabilistic one by transforming each conditional probability into conditional consonant BBA [18] that is discounted contextually [19] to spread beliefs on subsets (thus we obtain a "filled" matrix).

Smoothing. We use the credal γ variable (Eq. (20)) that combines both forward-backward algorithms to filter (offline) a noise BBA. Fig. 2-top shows the smoothing results with likelihoods (blue dots) vs. pignistic probabilities (red line) after application of the credal γ variable.

Inference. We apply the credal Viterbi algorithm on the previous set of likelihoods in order to decode the best sequences of states. The results of both the credal (Fig. 2-bottom-left) and probabilistic (Fig. 2-bottom-right) Viterbi decoding are shown. The proposed credal Viterbi scheme is able to detect the transitions (which size varies) as well as ambiguous instants (where the largest pignistic is low). These information are represented by green stems on the figure (with $\sigma = 1/N = 0.25$). The credal Viterbi yields better decoding here.

Classification. In order to assess the recognition criteria integrated in the Viterbi decoding (L_c^2), we propose a classification problem. We use 26 videos (about 3000 frames) of pole vaults analyzed by the three models (long jump (λ_1), high jump (λ_2) and pole vault (λ_3) models). We compare the criteria L_c^2 ($\lambda^* = \text{argmin}_{k=\{1,2,3\}} L_c^2(\lambda_k)$) to the usual log-likelihoods (L_l, $\lambda^* = \text{argmax}_{k=\{1,2,3\}} L_l(\lambda_k)$). We assess two things:

1) The classification rate: we compute it by dividing the number of correctly classified videos by 26. We obtain 81% for the CrHMM and 58% for the usual HMM. The classification rate is better for the CrHMM mainly because of doubt that is used to model knowledge about states. When a wrong decision is made in a HMM, it is propagated. In the credal Viterbi (CrHMM), the decision is not propagated since we keep the belief mass for the next instant.

2) The mean square of the difference between the two best normalized criteria values (by ranking the values for the criteria L_l and L_c^2): it reflects the "discriminative power" during classification (ideally one criteria equals 1 and the two others are null). We obtain 7% for the CrHMM and 1% for the usual HMM. This means that the CrHMM is much more discriminative than the HMM on this dataset therefore, the decision is more reliable.

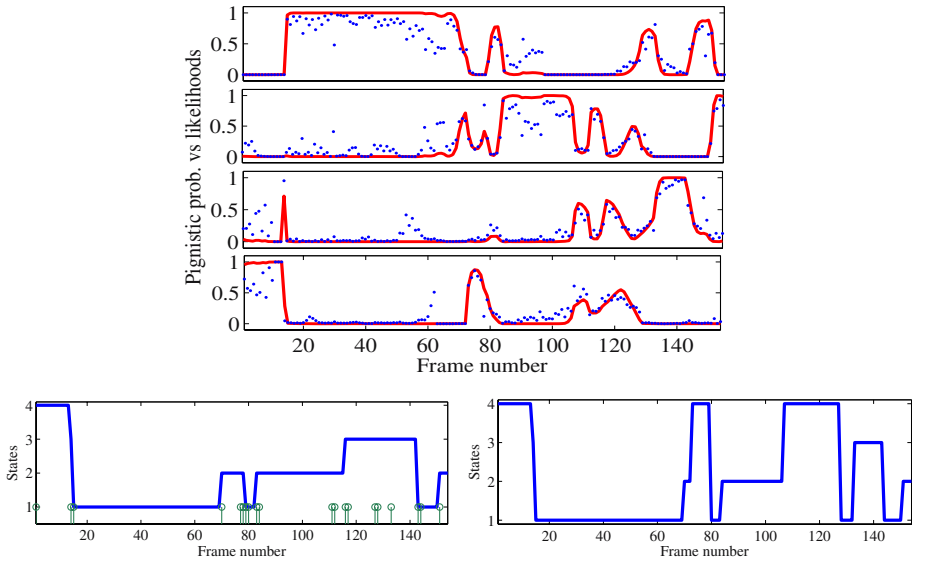


Fig. 2. Results: **(top)** Gaussian mixture outputs (blue dots) and smoothing by credal γ variable (red, bold). **(bottom-left)** Credal Viterbi decoding in CrHMM. **(bottom-right)** Viterbi decoding of HMM. Ground truth: running on $[12, 65]$, jumping on $[66, 103]$, falling on $[104, 154]$ (there is no standing-up action in this sequence).

8 Conclusion

We have presented a general methodology to use HMM machinery in Transferable Belief Model (TBM) framework which is useful when knowledge is represented by belief functions instead of probabilities. The proposed modelling exploits doubt and conflict emphasized in the TBM. Experiments have illustrated some concepts and demonstrated encouraging results for video-oriented applications. Future work will be focused on learning and comparison with fuzzy HMM.

References

1. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. of the IEEE 77, 257–285 (1989)
2. Murphy, K.P.: Dynamic Bayesian Networks: Representation, inference and learning. PhD thesis, UC Berkeley (CSD) (2002)
3. Shah, M.: Understanding human behavior from motion imagery. Machine Vision and Applications 14, 210–214 (2003)
4. Smets, P., Kennes, R.: The Transferable Belief Model. Artificial Intelligence 66(2), 191–234 (1994)
5. Ramasso, E., Pellerin, D., Rombaut, M.: Belief Scheduling for the recognition of human action sequence. In: Int. Conf. on Inf. Fusion, Florence, Italia (2006)

6. Pieczynski, W.: Multisensor triplet Markov chains and theory of evidence. IJAR 45, 1–16 (2007)
7. Smets, P.: Beliefs functions: The Disjunctive Rule of Combination and the Generalized Bayesian Theorem. IJAR 9, 1–35 (1993)
8. Xu, H., Smets, P.: Evidential reasoning with conditional belief functions. Uncertainty in Artificial Intelligence 94, 598–606 (1994)
9. Yaghlane, B.B., Smets, P., Mellouli, K.: Directed evidential networks with conditionnal belief functions. In: Nielsen, T.D., Zhang, N.L. (eds.) ECSQARU 2003. LNCS (LNAI), vol. 2711, pp. 291–305. Springer, Heidelberg (2003)
10. Yaghlane, B., Smets, P., Mellouli, K.: Belief Function Independence: II. The Conditional Case. IJAR 31, 31–75 (2000)
11. Dubois, D., Prade, H.: A set-theoretic view of belief functions – Logical operations and approximation by fuzzy sets. I. J. of Gen. Sys. 12(3), 193–226 (1986)
12. Delmotte, F., Smets, P.: Target identification based on the Transferable Belief Model interpretation of Dempster-Shafer model. IEEE TSMC 34(4), 457–471 (2004)
13. Ristic, B., Smets, Ph.: The TBM global distance measure for the association of uncertain combat ID declarations. Information Fusion 7, 276–284 (2006)
14. Denoeux, T., Yaghlane, A.B.: Approximating the combination of belief functions using the fast Moebius transform in a coarsened frame. IJAR 37, 77–101 (2002)
15. Ramasso, E., Panagiotakis, C., Rombaut, M., Pellerin, D.: Human action recognition in videos based on the Transferable Belief Model – Application to athletics jumps. Pattern Analysis and Applications (2007)
16. Panagiotakis, C., Ramasso, E., Tziritas, G., Rombaut, M., Pellerin, D.: Shape-motion based athlete tracking for multilevel action recognition. In: 4th Int. Conf. on Articulated Motion and Deformable Object, Spain, pp. 385–394 (2006)
17. Vannoorenberghe, P., Smets, Ph.: Partially supervised learning by a Credal EM approach. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 956–967. Springer, Heidelberg (2005)
18. Dubois, D., Prade, H., Smets, P.: New semantics for quantitative possibility theory. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, pp. 410–421. Springer, Heidelberg (2001)
19. Mercier, D., Quost, B., Denoeux, T.: Refined modeling of sensor reliability in the belief function framework using contextual discounting. Information Fusion (to appear)

Uncertainty in Semantic Ontology Mapping: An Evidential Approach

Najoua Laamari[1] and Boutheina Ben Yaghlane[2]

[1] LARODEC, Université de Tunis,
ISG Tunis Tunisia
laamarinajoua@yahoo.fr
[2] LARODEC, Université de Tunis,
IHEC Carthage Présidence 2016 Tunisia
boutheina.yaghlane@ihec.rnu.tn

Abstract. In this paper we propose a new tool called OWL-CM (OWL Combining Matcher) that deals with uncertainty inherent to ontology mapping process. On the one hand, OWL-CM uses the technique of similarity metrics to assess the equivalence between ontology entities and on the other hand, it incorporates belief functions theory into the mapping process in order to improve the effectiveness of the results computed by different matchers and to provide a generic framework for combining them. Our experiments which are carried out with the benchmark of Ontology Alignment Evaluation Initiative 2006 demonstrate good results.

1 Introduction

Semantic heterogeneity has been identified as one of the most important issue in information integration [5]. This research problem is due to semantic mismatches between models. Ontologies which provide a vocabulary for representing knowledge about a domain are frequently subjected to integration.

Ontology mapping is a fundamental operation towards resolving the semantic heterogeneity. It determines mappings between ontologies. These mappings catch semantic equivalence between ontologies. Experts try to establish mappings manually. However, manual reconciliation of semantics tends to be tedious, time consuming, error prone, expensive and therefore inefficient in dynamic environments, and what's more the introduction of the Semantic Web vision has underscored the need to make the ontology mapping process automatic.

Recently, a number of studies that are carried out towards automatic ontology mapping draw attention to the difficulty to make the operation fully automatic because of the cognitive complexity of the human. Thus, since the (semi-) automatic ontology mapping carries a degree of uncertainty, there is no guarantee that the outputted mapping of existing ontology mapping techniques is the exact one.

In this context, we propose a new tool called OWL-CM (OWL Combining Matcher) with the aim to show how handling uncertainty in ontology mapping process can improve effectiveness of the output. It uses the Dempster-Shafer

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 418–429, 2007.

theory of evidence [11] to deal with uncertainty inherent to the mapping process, especially when interpreting and combining the results returned by different matchers. Our approach is based on similarity metrics in order to assess the correspondence between ontology entities. We carried out experiments with the data sets of the Ontology Alignment Evaluation Initiative 2006 and preliminary tests show high precision and acceptable recall.

The rest of the paper is organized as follows. In section 2, we present a scope for the related works. Section 3 introduces the terminology material to our work. In section 4, we explain how evidential theory is applied by OWL-CM to handle uncertainty. Section 5 describes the architecture of our tool and the corresponding algorithm. Section 6 is devoted to experiments based on qualitative metrics. Section 7 concludes the paper and points out on some future directions.

2 Related Works

The approach that we propose is complementary to the methods that had been developed towards (semi-) automatic mapping between ontologies and database schemas. We are mainly based on four works. The first one [9] describes a domain specific multi-agent ontology mapping solution in the AQUA query answering system. It uses evidential formalism in order to incorporate uncertainty. This approach fits particularly for a query-answering scenario, where answer needs to be created in real time. As the context of our work is different from this one, the mapping approaches will be necessarily different. The second [1] presents a framework that uses evidential formalism for interpreting and combining results computed by several matchers. The third one [2] proposes a prototype for not full ontology mapping in the context of interaction between agents in an open and peer-to-peer environment. This prototype is based on the principles introduced in the previous framework. As opposed to Besana ([1], [2]), in our work we perform a *full mapping*. Finally, QOM [7] is an example of system that uses similarity metrics to compare features of ontology entities (e.g., their labels, subclasses, domains and ranges of properties, restrictions, etc.). QOM is not fit for a particular context like our tool. However, as opposed to our approach, it does not deal with uncertainty. It considers the similarity metric between ontological entities as a weighted combination of similarities of various features of entities. We also note that QOM considers both the quality of mapping results as well as the run-time complexity. However, we currently concentrate on the effectiveness of the mapping generation algorithm and leave the efficiency issue aside.

3 Preliminaries

In this section we draw up a list of preliminary concepts to use throughout the paper.

1. **Knowledge Model:** We define an *ontology* O as follows:

$$O := (C, H_c, R_c, R_d, H_rc, H_rd, I, R_i, A)$$

- C: Set of concepts (instances of 'owl:Class');
- H_c: Subsumption hierarchy of C (a binary relation: 'rdfs:subClassOf');
- R_c: Set of relations between single concepts (instances of 'owl:Object Property');
- R_d: Set of relations that relate single concepts to data types (instances of 'owl:DatatypeProperty');
- H_rc: Subsumption hierarchy of R_c ('rdfs:subPropertyOf');
- H_rd: Subsumption hierarchy of R_d ('rdfs:subPropertyOf');
- I: Sets of individuals (instances of classes c ∈ C);
- R_i: Set of property instances;
- A: Set of axioms that can be used to infer knowledge from already existing one.

2. **Candidate Mapping:** We define a *candidate mapping* as a pair of entities $({e_i}^1, {e_j}^2)$ that is not yet in map.

3. **Result Mapping:** We define a *result mapping* as a pair of entities that had been related, $\langle {e_i}^1, \equiv, {e_j}^2 \rangle$ denotes that entity ${e_i}^1$ is equivalent to entity ${e_j}^2$, whereas $\langle {e_i}^1, \perp, {e_j}^2 \rangle$ denotes that the two entities are not equivalent.

4. **Ontology Mapping Algorithm:** We define an *ontology mapping algorithm* as a partial function *map* that receives two ontologies O_1 and O_2 and returns a set M of result mappings.

$$
\begin{gathered}
map : O \times O \rightarrow M \\
M = \{\langle {e_i}^1, r, {e_j}^2 \rangle | {e_i}^1 \in O_1, {e_j}^2 \in O_2\} \\
r : e \times e \rightarrow \{\equiv, \perp\} \\
\text{with } \forall\ {e_i}^1 (\exists\ {e_j}^2 : map({e_i}^1, {e_j}^2) = \langle {e_i}^1, \equiv, {e_j}^2 \rangle\ \vee \\
\forall\ {e_j}^2 : map({e_i}^1, {e_j}^2) = \langle {e_i}^1, \perp, {e_j}^2 \rangle) \\
e \in \{C \cup R_c \cup R_d \cup I\}
\end{gathered}
$$

5. **Similarity Measure:** The *similarity measure*, *sim*, is a function defined in [3] based on the vocabularies ε_1 of the ontology O_1 and ε_2 of the ontology O_2 as follows:

$$sim\colon \varepsilon \times \varepsilon \times O \times O \rightarrow [0..1]$$

- *sim(a, b) = 1 ⇔ a = b: two objects are assumed to be identical.*
- *sim(a, b) = 0 ⇔ a ≠ b: two objects are assumed to be different and have no common characteristics.*
- *sim(a, a) = 1: similarity is reflexive.*
- *sim(a, b) = sim(b, a): similarity is symmetric.*
- *Similarity and distance are inverse to each other.*

A similarity measure function assesses the semantic correspondence between two entities based on some features. In table 1, we draw up the list of similarity measures employed depending on the type of entities to be mapped (some similarity measures definitions are described in appendix A). Furthermore, we distinguish between two types of similarity: the *syntactic one* assessed by

Table 1. Features and Measures for Similarity

Entities to be compared	No.	Feature (f)	Similarity measure
Concepts: C	1	(label, C1)	$sim_{strsim}(C1, C2)$
	2	(sound (ID), C1)	$sim_{streql}(C1, C2)$
	3	(label, C1)	$sim_{strsyn}(C1, C2)$
	4	(C1,equalTo, C2) relation	$sim_{expeql}(C1, C2)$
	5	(C1,inequalTo, C2) relation	$sim_{expineq}(C1, C2)$
	6	all (direct-sub-concepts, S1)	$sim_{setsim}(S1, S2)$
Relations: R_c	7	(sound (ID), R1)	$sim_{streql}(R1, R2)$
	8	(domain, R1)∧(range, R1)	$sim_{objeql}(R1, R2)$
	9	(domain, R1)∧(range, R1)	$sim_{objineq}(R1, R2)$
	10	all (direct-sub-properties, S1)	$sim_{setsim}(S1, S2)$
Relations: R_d	131	(sound (ID), R1)	$sim_{streql}(R1, R2)$
	12	(domain, R1)∧(range, R1)	$sim_{objeql}(R1, R2)\wedge$ $sim_{streql}(R1, R2)$
	13	(domain, R1)	$sim_{objineq}(R1, R2)$
	14	all (direct-sub-properties, S1)	$sim_{setsim}(S1, S2)$
Instances: I	15	(label, I1)	$sim_{strsim}(I1, I2)$
	16	(ID, I1)	$sim_{streql}(I1, I2)$
	17	(I1,equalTo, I2) relation	$sim_{expeql}(I1, I2)$
	18	(I1,inequalTo, I2) relation	$sim_{expineq}(I1, I2)$
	19	(direct-parent-concept, I1)	$sim_{objeql}(I1, I2)$

the measures that evaluate distance between strings (e.g., String similarity and String equality) and the other measures dedicated to assess *semantic similarity* (e.g., String synonymy, Explicit equality and Set similarity).

6. **SEE (Semantic Equivalent Entity):** Depending on the type of entities, we formally define the *semantic equivalence* between two entities as follows:

   Definition (SEE). An entity ${e_j}^2$ is semantically equivalent to an entity ${e_i}^1$ such that $({e_i}^1, {e_j}^2) \in \{C^1 \times C^2 \cup I^1 \times I^2\}$, i.e., $\langle {e_i}^1, \equiv, {e_j}^2 \rangle$, if at least one of the following conditions is true:
   $sim_{expeql}({e_i}^1, {e_j}^2) = 1$, or
   $\forall sim_k$, with $k \neq expeql, sim_k({e_i}^1, {e_j}^2) = 1$

   An entity ${e_j}^2$ is semantically equivalent to an entity ${e_i}^1$ such that $({e_i}^1, {e_j}^2) \in \{{R_c}^1 \times {R_c}^2 \cup {R_d}^1 \times {R_d}^2\}$, i.e., $\langle {e_i}^1, \equiv, {e_j}^2 \rangle$, if:
   $\forall sim_k, sim_k({e_i}^1, {e_j}^2) = 1$

7. **USEE (Uncertain Semantic Equivalent Entity):** We extend the definition of SEE to USEE in order to be used throughout the process of handling uncertainty when performing and combining matchers.

Definition (USEE). An entity that we said to be uncertain and semantically equivalent to an ontological entity $e \in O_1$ is a pair (Θ, m), where:
$\Theta = E$, $E \in \{C^2, {R_c}^2, {R_d}^2, I^2\}$
m is a belief mass function.

4 Handling Uncertainty

The Dempster-Shafer theory of evidence [11] presents some advantages that encourage us to choose among other theories. In particular, it can be used for the problems where the existing information is very fragmented, and so the information can not be modelled with a probabilistic formalism without making arbitrary hypotheses. It is also considered as a flexible modelling tool making it possible to handle different forms of uncertainty, mainly the ignorance. Moreover, this theory provides a method for combining the effect of different beliefs to establish a new global belief by using Dempster's rule of combination.

The belief mass function *m(.)* is the basic concept of this theory ([11], [12]). It assigns some belief mass in the interval [0,1] to each element of the power set 2^{Θ} of the frame of discernment Θ. The total mass distributed is 1 and the *closed world hypothesis* (i.e. $m(\emptyset) = 0$) is generally supported. In our work, $\Theta = \{{e_1}^2, \ldots, {e_n}^2\}$, where ${e_i}^2 \in O_2$. The letter Φ in table 2 is the set of all candidate mappings.

Table 2. Frame of Discernment and Candidate Mappings Set

	${e_1}^2$	...	${e_m}^2$	$\Rightarrow \Theta$
${e_1}^1$	$({e_1}^1, {e_1}^2)$	...	$({e_1}^1, {e_m}^2)$	Φ
...	...	...	...	
${e_n}^1$	$({e_n}^1, {e_1}^2)$	...	$({e_n}^1, {e_m}^2)$	

In order to discover USEEs, we use n functions called matchers ($matcher_k$)[1]. A matcher compared to a "witness" that brings evidence in favor or against an advanced hypothesis. Matchers produce USEEs in order to support uncertainty. Some matchers are reliable than others. This is reflected in the confidence that is assigned to each matcher. The confidence is expressed like the mass that is distributed to Θ. For instance, if $matcher_1$ has a confidence of .6, then the masses assigned to the subsets should be normalized to sum .6, and .4 should be always affected to Θ.

We use Dempster's rule of combination to aggregate the produced USEEs. Figure 1 illustrates the architecture that we propose to discover USEEs. In addition, this theory makes it possible to express total ignorance. For instance, if the set that contains the entities having the same sound as the entity in question is empty, then the matcher $matcher_2$ will return a belief mass function $m(\Theta) = 1$.

[1] The index k is the no. of the matcher in the table 1.

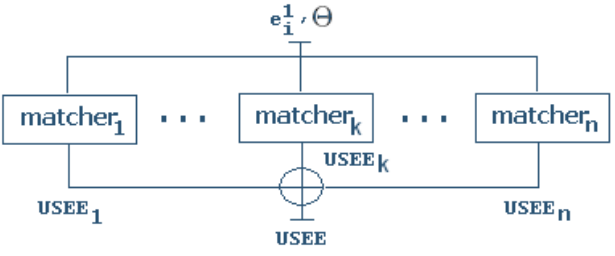


Fig. 1. Architecture for discovering USEEs

5 OWL-CM: Architecture and Algorithm

We give here the architecture of our proposed tool OWL-CM and the corresponding algorithm.

5.1 Architecture

The proposed architecture (see figure 2) contains four components. The *transformer* takes as input two ontologies (O_1 and O_2) and constructs for each one a database (DB_1 and DB_2). The database schema meets the standard schema that we designed based on some axioms of RDF(S) and OWL languages. The figure 3 is a part of the class diagram of the database. The *filters* decide on result mappings. Whereas *simple matchers* and *complex matchers* assess the equivalence between entities.

5.2 Algorithm

The algorithm of *ontology mapping* follows five steps (see figure 4). In this paper, we just describe the first step of pre-mapping and the second one which performs

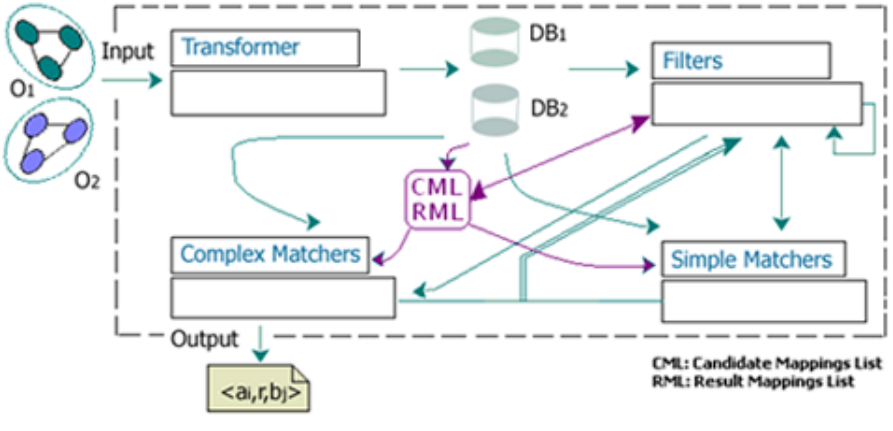


Fig. 2. Architecture of OWL-CM

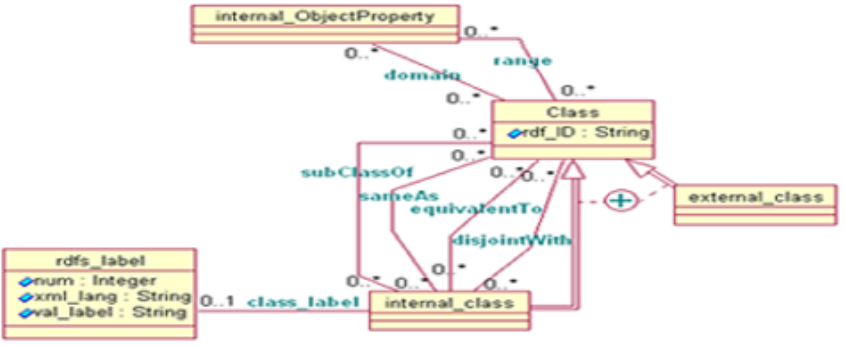


Fig. 3. Part of the Class Diagram of DB

the iteration about concepts mapping. The other ones will be described in incoming paper. The step of pre-mapping is devoted principally to convert each ontology formalism into a database by the transformer component. The first iteration of mapping performs alternately two tasks. One task uses some filters in order to screen the candidate mappings list and decide on the result mappings. The other task is devoted to the practicability of evaluating the similarity between concepts based on simple matchers and complex matchers.

Algorithm Ontology Mapping

A. Pre-mapping:
1. Convert O_1 into DB_1 and O_2 into DB_2.
2. $E2ML \leftarrow \emptyset$, $RML \leftarrow \emptyset$.

B. Iteration1: Concepts Mapping
1. Initialize the following variables:
   - $E2ML \leftarrow \{e2m_i s\}$, where each $e2m_i = \{({c_i}^1, {c_1}^2), \ldots, ({c_i}^1, {c_m}^2)\}$
2. Screening: For each $e2m_i \in E2ML$:
   2.1 Perform the filters f_1, f_2, f_3. In this case the filters return SEEs.
   2.2 Bring $e2m_i$ and RML up to date.
3. Uncertainty handling: For each $e2m_i \in E2ML$:
   3.1 Perform the simple matchers sm_1, sm_2, sm_3. Each sm_i return an $USEE_i$.
   3.2 Combine $USEE_1, USEE_2, USEE_3$. The result is a BMF1.
   3.3 Compute *bel* and *pl* for each $S_i \subseteq \Theta$ based on resulting belief mass function.
4. Screening: For each $e2m_i \in E2ML$:
   4.1 Perform the filters f_4, f_5.
   4.2 Bring $e2m_i$ and RML up to date.
5. Uncertainty handling: For each $e2m_i \in E2ML$ that was modified:
   5.1 Perform the complex matcher cm_1.
   5.2 Perform the simple matchers sm_1, sm_2, sm_3.

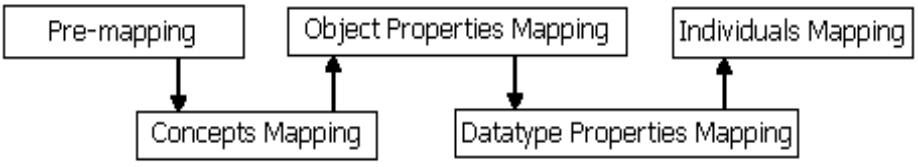


Fig. 4. Algorithm Steps

5.3 Combine $USEE_1$ of cm_1 and $USEE_1, USEE_2, USEE_3$ of sm_is. The result is a BMF2.

6. Screening: like in step 4.
7. Last screening: It is based on the following hypothesis: *If Θ of an $e2m_i$ is a singleton and this remaining concept $c_j{}^2$ isn't a member of any other frame of discernment of another $e2m_k$, then we suppose that this concept is the SEE of the concept $c_i{}^1$* which is in the process of being matched. For each $e2m_i \in E2ML$:
   7.1 Perform the filter f_6.
   7.2 Bring $e2m_i$ and RML up to date.
8. Iteration1 ending:
   4.1 Edit the result.
   4.2 Save a copy of the variable RML to be used during the following iterations.

6 Effectiveness Tests

The tests have been carried out with the data of the Ontology Alignment Evaluation Initiative 2006. We used data of the benchmark whose URL is:

http://oaei.ontologymatching.org/2006/benchmarks

The reference ontology is about bibliographic references. It contains 33 named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals. The complete ontology is that of the test 101. The ontologies are described in OWL-DL. In each test there are some information that have been retracted from the reference ontology. These various alterations makeit possible to evaluate the algorithm when some information are lacking. Table 3 presents what has been retracted from the reference ontology in some tests.

There are two categories of metrics to consider when evaluating an algorithm or a system. The first category evaluates the performance and is based on speed and space parameters. The second category evaluates the goodness of the algorithm output. Some of these metrics are derivatives of well-known metrics from the information retrieval domain [6]. Our experiments are restricted to the following metrics that belong to the second category:

Table 3. Ontologies of some tests

Test	Alterations in the reference ontology
101	No information has been discarded
102	Irrelevant ontology
103	It compares the ontology with its generalization in OWL Lite
104	It compares the ontology with its restriction in OWL Lite
201	No names (each label or identifier is replaced by a random one)
202	No names, no comment (comments have been suppressed as well)
203	No comments (there was misspelling)
204	Naming conventions are used for labels. Comments have been suppressed.
221	All subclass assertions to named classes are suppressed
223	The specialization hierarchy is expansed
230	Classes are flattened
240	Properties are suppressed and the specialization hierarchy is expansed

– *Precision*: The proportion of correct result mappings among those found:

$$Prec = \frac{TruePositives}{(TruePositives+FalsePositives)}$$

– *Recall*: The proportion of correct result mappings found:

$$Rec = \frac{TruePositives}{(TruePositives+FalseNegatives)}$$

– *FMeasure*: The harmonic mean of precision and recall:

$$FM = 2 \times \frac{(precision \times recall)}{(precision+recall)}$$

TruePositives is the number of correct result mappings contained in the answer list of the algorithm, *FalsePositives* is the number of incorrect result mappings an answer list contains, and *FalseNegatives* is the number of result mappings that the algorithm incorrectly predicted to be irrelevant.

6.1 Tests 101-104

Our results (see figure 5) show that our mapping algorithm enabled us to achieve 100% precision and 100% recall in the tests 101,103 and 104. The test 102 also shows the performance of the algorithm.

6.2 Tests 201-204

As the ontology 201 does not contain names and the ontology 202 contains neither names nor comments, we will not consider the results of these tests. In fact, our algorithm considers concept and property IDs as well as there labels therefore the only information that can be used to create these result mappings in the test 201 the "*rdfs:comment*" but our algorithm does not use it. Although the performed tests are not worth considering, even though they reveal a higher precision. The results of tests 203 and 204 (see figure 5) show that our algorithm creates the mapping with high precision for these tests. Recall values are also considerable.

6.3 Tests 221-247

Different categories of alteration regarding the specialization hierarchy, the properties, the instances, and/or the classes, have been carried out in each of these tests. The precision/recall rate of ontology mapping during these tests (see figure 5) is very high. This result confirms that our algorithm takes both syntactic and semantic similarity into account.

	Prec	Rec	FM
T101	1	1	1
T102	0	0	0
T103	1	1	1
T104	1	1	1
T201	1	0.10	0.19
T202	1	0.01	0.02
T203	1	1	1
T204	1	0.89	0.94

	Prec	Rec	FM
T221	1	1	1
T222	1	1	1
T223	1	0,99	0,99
T224	1	1	1
T225	1	1	1
T228	1	1	1
T230	1	1	1
T231	1	1	1
T232	1	1	1

	Prec	Rec	FM
T233	1	1	1
T236	1	1	1
T237	1	1	1
T238	1	0,99	0,99
T239	1	1	1
T240	1	0,97	0,98
T241	1	1	1
T246	1	1	1
T247	1	0,97	0,98

Fig. 5. Results of tests

6.4 Comments

Since the main goal of this work is to strengthen the precision of the ontology mapping with developing an approach that deals with uncertainty inherent to the mapping process, the means of the three metrics are encouraging (see table 4) compared to the ones resulted from the work of [9].

Concerning the benchmark, it served as an experiment bed to assess both strong and weak points of our algorithm and gives an idea of the prospects for improving the algorithm effectiveness. But, it does not present tests to interpret the use of some similarity measures based on the explicit assertions such as *explicit equality*.

We also draw attention to the fact that these tests assess only the effectiveness of the algorithm. Further, we will consider the run-time complexity.

7 Conclusion and Future Work

The ontology mapping is a core task towards interoperability between systems that use different ontologies. It addresses the problem of semantic heterogeneity

Table 4. Comparison between OWL-CM and DSSim

	Prec	Rec	FM
OWL-CM	0,9822	0,8528	0,9474
DSSim	0.98	0.55	0.7

between sources. This problem is getting worse in the Semantic Web where various issues which are linked to uncertainty in information systems occur. The results obtained with our algorithm turned out to be good, and compare positively with those obtained by other authors [9]. But the proposed work is still subject to improvements. In our future work we will implement the remaining three iterations of the algorithm. We will also investigate other possibilities that have been revealed during experiments and concerning the efficiency of our algorithm. Finally, we mention that there are some systems that worth studying in future work, such as Falcon-AO [8], HMatch [4], and OWL-CtxMatch [10].

References

1. Besana, P.: A framework for combining ontology and schema matchers with Dempster-Shafer. In: Proceedings of the International Workshop on Ontology Matching (OM-2006), Athens, Georgia, pp. 196–200 (2006a)
2. Besana, P.: Dynamic ontology mapping: a first prototype, SSP Talks, School of Informatics, University of Edinburgh (2006b)
3. Bisson, G.: Why and How to Define a Similarity Measure for Object Based Representation Systems. In: Towards Very Large Knowledge Bases, pp. 236–246. IOS Press, Amsterdam (1995)
4. Castano, S., Ferrara, A., Messa, G.: Results of the HMatch Ontology Matchmaker in OAEI 2006. In: Proceedings of the International Workshop on Ontology Matching (OM-2006), Athens, Georgia, pp. 134–143 (2006)
5. Convent,: Unsolvable problems related to the view integration approach. In: Atzeni, P., Ausiello, G. (eds.) ICDT 1986. LNCS, vol. 243, pp. 141–156. Springer, Heidelberg (1986)
6. Do, H., Melnik, S., Rahm, E.: Comparison of schema matching evaluations. In: Chaudhri, A.B., Jeckle, M., Rahm, E., Unland, R. (eds.) Web, Web-Services, and Database Systems. LNCS, vol. 2593, pp. 221–237. Springer, Heidelberg (2003)
7. Ehrig, M., Staab, S.: Qom - quick ontology mapping. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 289–303. Springer, Heidelberg (2004)
8. Hu, W., Cheng, G., Zheng, D., Zhong, X., Qu, Y.: The Results of Falcon-AO in the OAEI 2006 Campaign. In: Proceedings of the International Workshop on Ontology Matching (OM-2006), Athens, Georgia, pp. 124–133 (2006)
9. Nagy, M., Vargas-Vera, M., Motta, E.: Dssim-ontology mapping with uncertainty. In: Proceedings of the International Workshop on Ontology Matching (OM-2006), Athens, Georgia, pp. 115–123 (2006)
10. Niedbala, S.: OWL-CtxMatch. In: Proceedings of the International Workshop on Ontology Matching (OM-2006), Athens, Georgia, pp. 165–172 (2006)
11. Shafer, G.: A mathematical theory of evidence. Princeton University Press, Princeton (1976)
12. Smets, P.: Belief functions. In: non-standard logics for automated reasoning, pp. 253–286. Academic Press, London (1988)

Appendix A. Similarity Measures

The list includes some definitions presented in [7], and some others that we propose.

- ***String Similarity*** *measures the similarity of two strings on a scale from 0 to 1, based on Levenshtein's EditDistance (ed).*

$$sim_{strsim}(c,d) := max(0, \frac{min(|c|,|d|) - ed(c,d)}{min(|c|,|d|)})$$

- ***String Equality*** *is a strict measure to compare strings.* All characters (char (x) at position x) of the two strings have to be identical.

$$sim_{streql}(c,d) := \begin{cases} 1 \; c.char(i) = d.char(i), \forall i \in [0,|c|] \text{ with } |c| = |d|, \\ \\ 0 \;\; \text{otherwise.} \end{cases}$$

- ***String Synonymy*** checks if two strings are synonyms, based on WORDNET.

$$sim_{strsyn}(c,d) := \begin{cases} 1 \;\; \text{c and d are synonyms,} \\ \\ 0 \;\; \text{otherwise.} \end{cases}$$

- ***Explicit Equality*** checks whether a logical assertion already forces two entities to be equal. In an OWL ontology, this assertion is expressed by using the axiom "owl:sameAs". We refer to these assertions as "equalTo".

$$sim_{expeql}(a,b) := \begin{cases} 1 \; \exists \text{ assertion (a,"equalTo",b),} \\ \\ 0 \;\; \text{otherwise.} \end{cases}$$

- ***Set Similarity*** measures the similarity of two sets of entities. Given two sets E and F, it compares the entities e_i with all the entities f_j, and vice versa. The comparison is based on one or more similarity measures that are listed above.

$$sim_{setsim}(E,F) := \begin{cases} 1 \;\; \text{if } C_1, C_2, C_3, \text{ or } C_4, \\ \\ 0 \;\; \text{otherwise.} \end{cases}$$

C_1 : $(E = F) : |\{e_i \in F\}| = |\{f_j \in E\}| = |E| = |F|$;
C_2 : $(E \subset F) : |\{e_i \in F\}| = |E|$ and $|\{f_j \in E\}| \neq |F|$, and $\frac{|E|}{|F|} \geq .75$;
C_3 : $(E \supset F) : |\{e_i \in F\}| \neq |E|$ and $|\{f_j \in E\}| = |F|$, and $\frac{|F|}{|E|} \geq .75$;
C_4 : $(E \cap F \neq \emptyset) : |\{e_i \in F\}| \neq |E|$ and $|\{e_i \in F\}| \neq 0$ and $|\{f_j \in E\}| \neq |F|$, and $\frac{|\{e_i \in F\}|}{|F|} \geq .75$ and $\frac{|\{f_j \in E\}|}{|E|} \geq .75$.

Measures of Ruleset Quality Capable to Represent Uncertain Validity

Martin Holeňa

Institute of Computer Science, Academy of Sciences of the Czech Republic,
Pod vodárenskou věží 2, 18207 Praha 8, Czech Republic
martin@cs.cas.cz*

Abstract. The paper deals with quality measures of rules extracted from data, more precisely with measures of the whole extracted rulesets. Three particular approaches to extending ruleset quality measures from classification to general rulesets are discussed, and one of them, capable to represent uncertain validity of rulesets for objects, is elaborated in some detail. In particular, a generalization of ROC curves is proposed. The approach is illustrated on rulesets extracted with four important methods from the well-known iris data.

Keywords: rules extraction from data, quality measures, Boolean rules, fuzzy rules, observational logic, classification, ROC curves.

1 Introduction

Logical formulas of specific kinds, usually called *rules*, are a traditional way of formally representing knowledge. Therefore, it is not surprising that they are also the most frequent representation of the discovered knowledge in data mining. Existing methods for rules extraction are based on a broad variety of paradigms and theoretical principles. However, methods relying on different underlying assumptions can lead to the extraction of different or even contradictory rulesets from the same data. Moreover, the set of rules extracted with a particular method can substantially depend on some tunable parameter or parameters of the method, such as significance level, thresholds, size parameters, tradeoff coefficients etc. For that reason, it is desirable to have measures of various qualitative aspects of the extracted rulesets. So far, such measures are available only for sets of classification rules, and their dependence on tunable parameters can be described only for classification into two classes [1,2]. As far as more general kinds of rules are concerned, measures of quality have been proposed only for individual rules [3,4,5,6,7], or for contrast sets of rules, which finally can be replaced with a single rule [8,9]; if a whole ruleset is taken into consideration, then only as a context for measuring the quality of an individual rule [10,11].

* The research reported in this paper has been supported by the grant 201/05/0325 of the Grant Agency of Czech Republic, and partially supported by the institutional research plan AV0Z10300504. Highly appreciated is the contribution of Vojta Hlaveš, who has run all the tests.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 430–442, 2007.

The present paper notices that the existing quality measures for sets of classification rules can be extended in various ways to general rulesets and focuses on extensions capable to represent uncertain validity of rulesets for particular objects. The proposed extensions are introduced in Section 3, after the basic typology and important examples of rules extraction methods are recalled in the next section, and before the dependence on method parameters, treated in Section 4. The paper concludes with an illustration on the well-known iris data.

2 Methods for the Extraction of Rules from Data

2.1 Typology of Rules Extraction Methods

The most natural base for differentiating between existing rules extraction methods is the *syntax and semantics of the extracted rules*. Syntactical differences between them are, however, not very deep since principally, any rule r from a ruleset $\mathcal{R}$ has one of the forms $S_r \sim S'_r$, or $A_r \rightarrow C_r$, where S_r, S'_r, A_r and C_r are formulas of the considered logic, and $\sim$, $\rightarrow$ are symbols of its language, defined in such a way that $S_r \sim S'_r$ is symmetric with respect to S_r, S'_r in the sense that its validity always coincides with that of $S_r \sim S'_r$ whereas $A_r \rightarrow C_r$ is not symmetric with respect to A_r, C_r in that sense. More precisely, $\sim$ and $\rightarrow$ are the connectives equivalence ($\equiv$) and implication, respectively, in the case of a propositional logic, whereas they are generalized quantifiers in the case of a predicate logic. To distinguish the involved formulas in the asymmetric case, A_r is called *antecedent* and C_r *consequent* of r.

The more important is the semantic of the rueles (cf. [5]), especially the difference is between *rules of the Boolean logic* and *rules of a fuzzy logic*. Due to the semantics of Boolean and fuzzy formulas, the former are valid for crisp sets of objects, whereas the validity of the latter is a fuzzy set on the universe of all considered objects. Boolean rulesets are extracted more frequently, especially some specific types of them, such as *classification rulesets* [1,6]. These are sets of asymmetric rules such that $(A_r)_{r\in\mathcal{R}}$ and $\{C_r\}_{r\in\mathcal{R}}$ partition the set $\mathcal{O}$ of considered objects, where $\{C_r\}_{r\in\mathcal{R}}$ stands for the distinct formulas in $(C_r)_{r\in\mathcal{R}}$. Abandoning the requirement that $(A_r)_{r\in\mathcal{R}}$ partitions $\mathcal{O}$ (at least in the crisp sense) allows to generalize those rulesets also to fuzzy antecedents. For Boolean antecedents, however, this requireme nt entails a natural definition of the validity of the whole ruleset $\mathcal{R}$ for an object x. Assuming that all information about x conveyed by $\mathcal{R}$ is conveyed by the rule r covering x (i.e., with A_r valid for x), the validity of $\mathcal{R}$ for x can be defined to coincide with the validity of C_r for x.

As far as the Boolean predicate logic is concerned, generalized quantifiers both for symmetric and for asymmetric rules were studied in the 1970s within the framework of the *observational logic* [12]. In that logic, the truth function Tf_Q of a generalized quantifier Q is solely a function of the four-fold table

$$\begin{array}{cc|cc} & & S'_r & \neg S'_r \\ & & C_r & \neg C_r \\ \hline S_r & A_r & a & b \\ \neg S_r & \neg A_r & c & d \end{array}. \tag{1}$$

Hence, Tf_Q is a $\{0,1\}$-valued function on quadruples of nonnegative integers. For symmetric rules, *associational quantifiers* are used, defined through the condition

$$a' \geq a \;\&\; b' \leq b \;\&\; c' \leq c \;\&\; d' \geq d \;\&\; \mathrm{Tf}_Q(a,b,c,d) = 1 \rightarrow \mathrm{Tf}_Q(a',b',c',d') = 1. \tag{2}$$

For asymmetric rules, the more specific *implicational quantifiers* are used, which are defined through the stronger condition

$$a' \geq a \;\&\; b' \leq b \;\&\; \mathrm{Tf}_Q(a,b,c,d) = 1 \rightarrow \mathrm{Tf}_Q(a',b',c',d') = 1. \tag{3}$$

Observe that this condition covers the frequently encountered *association rules* [5,13,14,15] (since they have been proposed independently of observational logic, the terminology is a bit confusing here: although associational rules are asymmetric, their name evokes the quantifier for the symmetric ones).

Orthogonally to the typology according to the semantics of the extracted rules, all extraction methods can be divided into two large groups:

- Methods that extract logical rules from data *directly*, without any intermediate formal representation of the discovered knowledge. Such methods have always formed the mainstream of the extraction of Boolean rules: from the observational logic methods [12] and the method AQ [16,17] in the late 1970s, through the extraction of association rules [13,14,15] and the method *CN2* [18], relying on a paradigm similar to that of AQ, to the recent methods based on *inductive logic programming* [19,20] and *genetic algorithms* [21]. They include also important methods for fuzzy rules, in particular *ANFIS* [22,23] and *NEFCLASS* [24,25], *fuzzy generalizations* of *observational logic* [26,27] and a recent method based on *fuzzy transform* [28].
- Methods that employ some *intermediate representation* of the extracted knowledge, useful by itself. This group includes two important kinds of methods: *classification trees* [29,30] and methods based on *artificial neural networks (ANN)*. The latter are used both for Boolean and for fuzzy rules [32,33,34] (cf. also the survey papers [35,36]).

2.2 Important Examples of Rules Extraction Methods

In this subsection, the basic principles of four important rules extraction methods will be recalled. Their choice attempts to reflect the various aspects of the differences within the spectrum of the existing methods. In particular:

- methods 1.-3. extract Boolean rules, method 4. fuzzy rules;
- among the Boolean methods, 1. extracts classification rules, 2. predicate rules with an associational quantifier, and 3. with an implicational quantifier;
- 2. and 3. are direct methods without an intermediate representation, 1. is a classification tree method, and 4. is ANN-based.

1. The method *CART* [29] recursively partitions data with axis-orthogonal hyperplanes, where the choice between different partitions relies on some impurity index, based on estimates $\hat{p}(c|v)$ of the conditional probability that an object in the vertex v belongs to the class c. For testing, the implementation of CART in MATLAB has been used, with the impurity index being either the Gini index $\sum_{c \neq c'} \hat{p}(c|v)\hat{p}(c'|v)$, or the deviance $-\sum \hat{p}(c|v) \ln \hat{p}(c|v)$.
2. The *Fisher quantifier* $\sim_\alpha^F$, $\alpha \in (0,1)$ has its truth function $\mathrm{Tf}_{\sim_\alpha^F}$ defined in such a way that the rule $S_r \sim_\alpha^F S'_r$ is valid exactly for those data for which statistical testing of the null hypothesis of independence of A_r and C_r against the alternative of their positive dependence with the one-tailed Fisher exact test leads to rejecting the null hypothesis on the significance level α [12]. Hence, $\mathrm{Tf}_{\sim_\alpha^F}(a,b,c,d) = 1$ iff $ad > bc$ & $\sum_{i=a}^{a+\min(b,c)} \frac{\binom{a+c}{i}\binom{b+d}{a+b-i}}{\binom{a+b+c+d}{a+b}} \leq \alpha$. For testing, the implementation in the LISP-Miner system [37] was used.
3. The quantifier *founded implication* $\rightarrow_{s,\theta}$, $s, \theta \in (0,1]$ has its truth function $\mathrm{Tf}_{\rightarrow_{s,\theta}}$ defined in such a way that the rule $A_r \rightarrow_{s,\theta} C_r$ is valid exactly for those data for which the conditional probability $p(C_r|A_r)$ of the validity of C_r conditioned on A_r, estimated with the unbiased estimate $\frac{a}{a+b}$, is at least θ, whereas A_r and C_r are simultaneously valid in at least the proportion s of the data [12]. Hence, $\mathrm{Tf}_{\rightarrow_{s,\theta}} = 1$ iff $\frac{a}{a+b} \geq \theta$ & $\frac{a}{a+b+c+d} \geq s$. As was pointed out in [38], rules with this quantifier are actually association rules with support s and confidence θ. Also in this case, the implementation in LISP-Miner was used for testing.
4. An ANN-based method for the extraction of rules of any fuzzy propositional logic that was proposed in [39]. It extracts always a single rule $S_r \equiv S'_r$ with atomic S'_r and S_r in disjunctive normal form (DNF), each atom of which contains a single object variable modelled with a finitely-parameterized fuzzy set (e.g., Gaussian, triangular, sigmoid). The architecture of the ANN reflects the construction of the S_r. An example output of such ANN is depicted in Figure 1. For testing, the method has been implemented in MATLAB [39].

3 Ruleset Quality Measures – Classification and Behind

3.1 Existing Measures of Quality for Classification Rulesets

A survey of measures for classification rulesets (with possibly fuzzy antecedents) has been given in the monograph [1]. Space limitation allows to recall here only the main representatives of the first two from the 4 classes of measures considered there, neglecting the other classes, inseparability and resemblance:

1. *Inaccuracy* measures the discrepancy between the true class of the considered objects and the class predicted by the ruleset. Its most frequently encountered representative is the *quadratic score* (also called Brier score):

$$\mathrm{Inacc} = \frac{1}{|\mathcal{O}|} \sum_{x \in \mathcal{O}} \sum_{C \in \{C_r\}_{r \in \mathcal{R}}} \left(\delta_C(x) - \hat{\delta}_C(x) \right)^2, \tag{4}$$

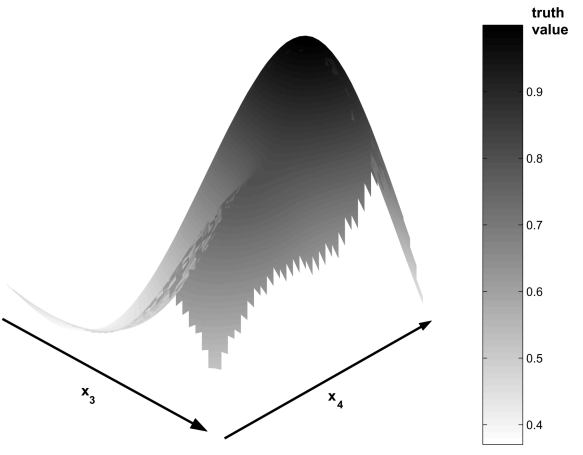


Fig. 1. A 2-dimensional cut for the dimensions x_3 and x_4 of the graph of a mapping computed by a neural network with 12 input neurons, 5 hidden neurons and 1 output neuron, each input of which corresponds to a variable modelled with a Gaussian fuzzy set, whereas the output returns the truth grade of the equivalent formulas

where $|\ |$ denotes cardinality, $\mathcal{O}$ is the considered set of objects, $\delta_C(x) \in \{0,1\}$ is the validity of the proposition C for $x \in \mathcal{O}$, and $\hat{\delta}_C(x)$ is the agreement between C and the class predicted for x by $\mathcal{R}$. Hence, $\hat{\delta}_C(x) = \max_{C_r=C} \|A_r\|_x$, where $\|A_r\|_x \in \langle 0,1\rangle$ is the truth grade of A_r for x in the considered logic.

2. *Imprecision* measures the discrepancy between the probability distribution of the classes, conditioned on the values of attributes occurring in antecedents, and the class predicted by the ruleset. Its most common representative is

$$\mathrm{Impr} = \frac{1}{|\mathcal{O}|} \sum_{x\in\mathcal{O}} \sum_{C\in\{C_r\}_{r\in\mathcal{R}}} \Big(\delta_C(x) - \hat{\delta}_C(x)\Big) \Big(1 - \hat{\delta}_C(x)\Big)^2 . \tag{5}$$

3.2 Extensions to More General Kinds of Rules

In the particular case of classification rulesets with Boolean antecedents, some algebra allows to substantially simplify (4)–(5):

$$\mathrm{Inacc} = \frac{2|\mathcal{O}^-|}{|\mathcal{O}|} = 1 - \frac{|\mathcal{O}^+| - |\mathcal{O}^-|}{|\mathcal{O}|},\ \mathrm{Impr} = \frac{|\mathcal{O}^-|}{|\mathcal{O}|} = 1 - \frac{|\mathcal{O}^+|}{|\mathcal{O}|},\ \text{where} \tag{6}$$

$$\mathcal{O}^+ = \{x \in \mathcal{O} : \mathcal{R} \text{ is valid for } x\},\ \mathcal{O}^- = \{x \in \mathcal{O} : \mathcal{R} \text{ is not valid for } x\}. \tag{7}$$

This not only shows that, in the case of Boolean antecedents, the quadratic score is sufficient to describe also the imprecision, but also suggests an approach how to extend those measures to general rulesets: to use (6)–(7) as the definition of measures (4)–(5). More generally, any measure of quality of classification rulesets with Boolean antecedents (e.g., any measure surveyed in [1]) that can

be reformulated by means of $\mathcal{O}^+$ and $\mathcal{O}^-$, can be extended in such a way that the reformulation is used as the definition of that measure for general rulesets.

For sets of asymmetric rules, also the notion of covering an object by a rule, which was recalled in Section 2, can be generalized. Notice, however, that for fuzzy antecedents, the validity of A_r, $r \in \mathcal{R}$ is a fuzzy set on $\mathcal{O}$. Consequently, the set $\mathcal{O}_\mathcal{R}$ of objects covered by $\mathcal{R}$ is a fuzzy set on $\mathcal{O}$, with membership function

$$\mu_\mathcal{R}(x) = \|(\exists r \in \mathcal{R})\ A_r\|_x = \max_{r \in \mathcal{R}} \|A_r\|_x. \tag{8}$$

Therefore, various generalizations of classification measures to general rulesets of asymmetric rules are possible: wherever $\mathcal{O}$ occurs in the definition of a measure for classification rulesets, either $\mathcal{O}$ or $\mathcal{O}_\mathcal{R}$ can occur in its general definition, provided $\mathcal{O}_\mathcal{R} \neq \emptyset$. To allow unified treatment of symmetric and asymmetric rules, $\mathcal{O}_\mathcal{R} = \mathcal{O}$ will be formally defined for the former. Observe that due to (8), $\mathcal{O}_\mathcal{R} = \mathcal{O}$ holds also for classification rulesets with Boolean antecedents.

A key feature of this approach is a multiple possible definition of validity of a general ruleset for an object. Indeed, any of the following definitions has its own justification, and coincides with the one given in Section 2 for classification rulesets with Boolean antecedents, thus being its correct extension:

(i) If the validity of $\mathcal{R}$ for x is viewed as the simultaneous validity of all rules, which in the case of fuzzy rules can be generalized to validity in at least a prescribed truth grade $t_{\min}$. Hence, $\mathcal{R}$ is valid for x iff $(\forall r \in \mathcal{R})$ r is valid for x, or more generally, iff $(\forall r \in \mathcal{R})$ $\|r\|_x \geq t_{\min}$. Notice that for individual objects, only the validity according to a propositional logic can be considered (since the validity of quantifiers depends on the set $\mathcal{O}$ as a whole).

(ii) If the validity of $\mathcal{R}$ for x is viewed as the validity of most of its rules, provided x is covered by $\mathcal{R}$:

$$\mathcal{R} \text{ is valid for } x \text{ iff } \mu_\mathcal{R}(x)|\{r \in \mathcal{R} : r \text{ is valid for } x\}| > \\ > \mu_\mathcal{R}(x)|\{r \in \mathcal{R} : \neg r \text{ is valid for } x\}|. \tag{9}$$

Notice that in terms of truth grades, the condition for the validity of $\mathcal{R}$ for x reads $\sum_{r \in \mathcal{R}} \|S_r \equiv S'_r\|_x > \sum_{r \in \mathcal{R}} \|\neg(S_r \equiv S'_r)\|_x$ for symmetric rules, and $\mu_\mathcal{R}(x) \sum_{r \in \mathcal{R}} \|A_r \to C_r\|_x > \mu_\mathcal{R}(x) \sum_{r \in \mathcal{R}} \|\neg(A_r \to C_r)\|_x$ for asymmetric rules. Moreover, (9) implies that $\mathcal{R}$ is not valid for x with $\mu_\mathcal{R}(x) = 0$.

(iii) If the validity of $\mathcal{R}$ for x is viewed, provided x is covered by $\mathcal{R}$, as uncertain, i.e., as a value from $\langle 0, 1\rangle$, the sets $\mathcal{O}^+$ and $\mathcal{O}^-$ from (7) turn to fuzzy sets on $\mathcal{O}$ with membership grades μ_+ and μ_-, respectively, such that

$$\mu_+(x) = \frac{|\{r \in \mathcal{R} : r \text{ is valid for } x\}|}{|\mathcal{R}|} \text{ if } \mu_\mathcal{R}(x) > 0, \tag{10}$$

$$\mu_-(x) = \frac{|\{r \in \mathcal{R} : \neg r \text{ is valid for } x\}|}{|\mathcal{R}|} \text{ if } \mu_\mathcal{R}(x) > 0, \tag{11}$$

adopting $\mu_+(x) = 0$, $\mu_-(x) = 1$ if $\mu_\mathcal{R}(x) = 0$ for compatibility with (ii).

In the sequel, the paper focuses on measures capable to represent the uncertain validity of rulesets, thus defined according to (iii). For them, the cardinalities used in the definitions are $|\mathcal{O}^{+/-}| = \sum_{x\in\mathcal{O}} \mu_{+/-}(x)$. For example, the measure

$$\mathrm{Inacc} = 1 - \frac{\sum_{x\in\mathcal{O}} (\mu_+(x) - \mu_-(x))}{|\mathcal{O}|} \tag{12}$$

is a generalization of (4), whereas the measures

$$\mathrm{Impr}_1 = 1 - \frac{\sum_{x\in\mathcal{O}} \mu_+(x)}{|\mathcal{O}|},\ \mathrm{Impr}_2 = 1 - \frac{\sum_{x\in\mathcal{O}} \mu_+(x)}{|\mathcal{O}_\mathcal{R}|} = 1 - \frac{\sum_{x\in\mathcal{O}} \mu_+(x)}{\sum_{x\in\mathcal{O}} \mu_\mathcal{R}(x)} \tag{13}$$

are generalizations of (5). Observe that for symmetric rules, as well as for classification rulesets with Boolean antecedents, $\mathcal{O}_\mathcal{R} = \mathcal{O}$ implies $\mathrm{Impr}_2 = \mathrm{Impr}_1$. Moreover, if the considered logic has the involutive negation, e.g., the Boolean, Łukasiewicz or product-Łukasiewicz logic (in applications, the choice of a fuzzy logic is often restricted to the involutive negation), then in addition $|\mathcal{O}^+|+|\mathcal{O}^-| = |\mathcal{O}_\mathcal{R}|$, thus only one of the measures Inacc and Impr_1 is needed, like in (6).

4 Influence of Method Parameters on Ruleset Quality

4.1 ROC Curves for Two-Class Classification

The rulesets that a particular method extracts from given data can substantially depend on values of various parameters of the method, such as tree depth or size for the CART method, significance level for the Fisher quantifier, support and confidence for association rules, or the number of hidden neurons and parameters of the input fuzzy sets for the ANN-based method proposed in [39]. Then also the results of applying quality measures to the ruleset depend on those parameter values. So far, the influence of parameter values has been systematically studied only for dichotomous classification when $\mathcal{R} = \{A \to C, \neg A \to \neg C\}$. In that case, putting $A_r = A, C_r = C$ allows the information about the validity of A and C for $\mathcal{O}$ to be again summarized by means of the four-fold table (1), which also depends on the parameter values. The influence of the parameter values on the result of dichotomous classification is usually investigated by means of the measures *sensitivity* $= \frac{a}{a+c}$ and *specificity* $= \frac{d}{b+d}$ [1]. Connecting points (1-specificity,sensitivity) $= (\frac{b}{b+d}, \frac{a}{a+c})$ for the considered parameter values forms a curve with graph in the unit square, called *receiver operating characteristic* (ROC), due to the area where such curves have first been in routine use. In machine learning, a modified version of those curves has been proposed, in which the points connected for considered parameter values are (b, a) [2]. The graph of

such a curve then lies in the rectangle with vertices $(0,0)$ and $(b+d, a+c)$, and is called *NP-graph*, due to the employed notation $P = a + c, N = b + d$.

The graphs of ROC curves and NP-graphs can provide information about the influence of parameter values not only on the sensitivity and specificity, but also on other measures. It is sufficient to complement the graph with isolines of the measure and to investigate their intersections with the original curve [2].

4.2 Extensions of ROC Curves to More General Kinds of Rules

Observe that the information about the validity of $\mathcal{R}$ for objects $x \in \mathcal{O}$ can be also viewed as information about the validity of a ruleset $\mathcal{R}' = \{A \to C\}$. However, $\mathcal{R}'$ is not any more a classification ruleset, but only a general one, which can be described only by means of the above introduced sets $\mathcal{O}_R$, $\mathcal{O}^+$, $\mathcal{O}^-$. In particular, $|\mathcal{O}^+| = a$ and $|\mathcal{O}^-| = b$, which suggests the possibility to generalize NP-graphs to general rulesets by means of a curve connecting points $(|\mathcal{O}^-|, |\mathcal{O}^+|)$ for the considered values of method parameters. For a generalization of ROC curves to general rulesets, those points have to be scaled to the unit square. Since the resulting curve will be used to investigate the dependence on parameter values, the scaling factor itself must be independent of those values. The only available factor fulfilling this condition is the number of objects, $|\mathcal{O}|$ (the other available factors, $|\mathcal{O}_R|$, $|\mathcal{O}^+|$ and $|\mathcal{O}^-|$ depend on the evaluations $\|S_r\|$ and $\|S'_r\|$, or $\|A_r\|$ and $\|C_r\|$, which in turn depend on the parameter values). Consequently, the proposed generalization of ROC curves will connect points $(\frac{|\mathcal{O}^-|}{|\mathcal{O}|}, \frac{|\mathcal{O}^+|}{|\mathcal{O}|})$.

It can be shown, using general results valid for any fuzzy logic [40], that whatever fuzzy logic is used in connection with the definition of the validity of $\mathcal{R}$ for $x \in \mathcal{O}$ according to (iii) on page 435, the points $(\frac{|\mathcal{O}^-|}{|\mathcal{O}|}, \frac{|\mathcal{O}^+|}{|\mathcal{O}|})$ always lie below the diagonal $([0,1],[1,0])$ (which is also trivially true if the definitions (i) or (ii) are used). This is depicted in Figure 2, together with isolines of the three example measures introduced in (12)–(13). Observe that the isolines of Impr_2 depend on the relationship between the cardinalities $|\mathcal{O}^+| = \sum_{x \in \mathcal{O}} \mu_+(x)$,

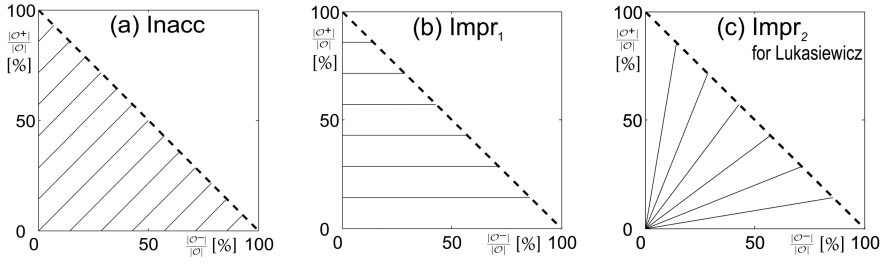


Fig. 2. Isolines of the measures introduced in (12)–(13), drawn with respect to the coordinates $(\frac{|\mathcal{O}^-|}{|\mathcal{O}|}, \frac{|\mathcal{O}^+|}{|\mathcal{O}|})$ of points forming the proposed generalization of ROC curves

$|\mathcal{O}^-| = \sum_{x\in\mathcal{O}} \mu_-(x)$ and $|\mathcal{O}_R| = \sum_{x\in\mathcal{O}} \mu_{\mathcal{R}}(x)$. The isolines depicted in Figure 2(c) correspond to the relationship $|\mathcal{O}_R| = |\mathcal{O}^+| + |\mathcal{O}^-|$, which is true in Łukasiewicz logic (thus in particular also in Boolean logic).

5 Illustration Using Fisher Iris Data

The proposed approach has been so far tested for six rules extraction methods (including the four recalled in Section 2) on three benchmark data sets, as well as on data from one real-world data mining task [41]. Here, it will be illustrated with some results obtained for the best known benchmark set, the iris data, originally used in 1930s by R.A. Fisher [42].

As to the methods from Section 2, the method CART has been used with trees of 2–6 leaves, each combined with Gini index and deviance, the Fisher quantifier with 5 significance levels, the founded implication with combinations of 10 values of s and 7 values of θ, and the ANN-based method from [39] with combinations of 2–4 hidden neurons and 3 particular fuzzy sets modelling input variables, each of them interpreted in Łukasiewicz and in product-Łukasiewicz logic. All methods were used to extract traditional rules for the iris data, concerning relationships between the values of the descriptive attributes (length and width of petals and sepals) and the kind of iris. In addition, the Fisher quantifier has been used to extract also rules concerning relationships between the values of different descriptive attributes. For the split of the data into training and test set, 10-fold cross validation was employed. Consequently, altogether $10 * (2 * 5 + 2 * 5 + 10 * 7 + 2 * 3 * 3) = 1080$ rulesets were extracted with those methods from the iris data. The number of rules in a CART ruleset equals the number of leaves of the tree, and always one DNF rule is extracted with the ANN-based method. On

Table 1. Results obtained for the rulesets extracted from the iris data, averaged over a 10-fold cross validation

CART		Gini index					deviance				
leaves		2	3	4	5	6	2	3	4	5	6
Impr_1		0.40	0.08	0.04	0.02	0.16	0.40	0.08	0.04	0.02	0.16
Fisher quantifier		rules as in other methods					only descriptive attributes				
significance level α		0.1%	0.5%	1%	5%	10%	0.1%	0.5%	1%	5%	10%
rules		77.4	104.6	111.8	151	158	159.2	228.6	277.6	427.6	520.8
Impr_1		0.24	0.25	0.26	0.27	0.27	0.21	0.21	0.22	0.22	0.22
founded implication		rules	Inacc	Impr_1	Impr_2	$s = 0.09$		rules	Inacc	Impr_1	Impr_2
$s =$	$\theta = 0.8$	71.7	0.12	0.06	0.06	$\theta = 0.8$		29.3	0.11	0.07	0.04
0.05	$\theta = 0.9$	59.8	0.09	0.05	0.04	$\theta = 0.9$		25.7	0.19	0.15	0.04
ANN-based method [39]		Gaussian input			triangular input			sigmoid input			
hidden neurons		2	3	4	2	3	4	2	3	4	
Impr_1: Łukasiewicz logic		0.04	0.05	0.04	0.20	0.20	0.18	0.30	0.25	0.25	
product-Łukasiewicz logic		0.69	0.69	0.69	0.11	0.18	0.18	0.69	0.69	0.68	

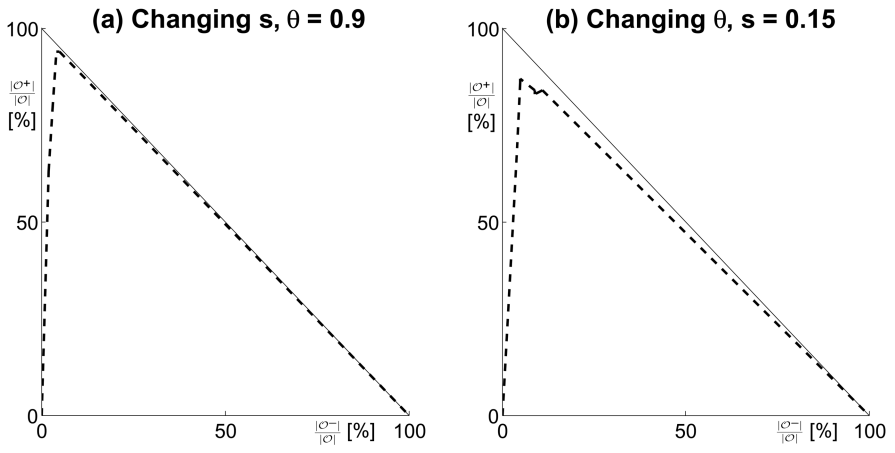


Fig. 3. Generalized ROC curves for rulesets extracted from the iris data by means of the founded implication: (a) changing s, $\theta = 0.9$; (b) changing θ, $s = 0.15$

the other hand, the size of rulesets extracted using the Fisher quantifier or the founded implication can vary considerably (Table 1).

Results obtained when applying the measures introduced in (12)–(13) to the extracted rulesets are given in Table 1. Since $\mathrm{Impr}_2 = \mathrm{Impr}_1$ for CART, the Fisher quantifier and the ANN-based method, and since always logics with the involutive negation have been used, solely Impr_1 has been computed for those methods. For the founded implication, only 4 from the 70 combinations of values of s and θ are given in the table, due to space limitations. Apart from confirming the expectation that the most precise classification to the kinds of iris is achieved with a specific classification method, the results show that the choice of values of method parameters has a much greater impact on the ruleset quality than the choice of the method itself. Whereas even methods relying on quite different theoretical principles yielded rulesets of comparable quality, inappropriate values of parameters turned the method from a useful one to a quite useless one. Finally, two examples of the proposed generalization of ROC curves for rulesets extracted by means of the founded implication are shown in Figure 3. According to Section 4, combining these curves with the isolines of Inacc, Impr_1 and Impr_2 allows to investigate the dependence of those measures on the values of the parameters s and θ (for a fixed value of the other parameter).

6 Conclusions

The paper has dealt with quality measures of rules extracted from data, though not in the usual context of individual rules, but in the context of whole rulesets. Three kinds of extensions of measures already in use for classification rulesets have been discussed, and one of them, capable to represent the uncertain validity

of general rulesets, has been elaborated in some detail. In particular, the concept of ROC-curves has been generalized, to enable investigating the dependence of general rulesets on the values of parameters of the extraction method.

The extent of the paper did not allow more then only to sketch the basic ideas of the proposed approach. However, the approach has already been tested on rulesets extracted from three benchmark and one real-world data sets by means of six methods attempting to cover a possibly broad spectrum of rules extraction methods. The results of those tests, a small sample of which has been presented in the last section, indicate that the approach is feasible and can contribute to the ultimate objective of quality measures: to allow comparing the knowledge extracted with different data mining methods and investigating how the extracted knowledge depends on the values of their parameters.

References

1. Hand, D.: Construction and Assessment of Classification Rules. John Wiley and Sons, New York (1997)
2. Fürnkranz, J., Flach, P.: ROC 'n' rule learning – towards a better understanding of covering algorithms. Machine Learning 58, 39–77 (2005)
3. Kaufman, K., Michalski, R.: An adjustable description quality measure for pattern discovery using the AQ methodology. Journal of Intelligent Information Systems 14, 199–216 (2000)
4. McGarry, K.: A survey of interestingness measures for knowledge discovery. Knowledge Engineering Review 20, 39–61 (2005)
5. Dubois, D., Hüllermeier, Prade, H.: A systematic approach to the assessment of fuzzy association rules. Data Mining and Knowledge Discovery 13, 167–192 (2006)
6. Geng, L., Hamilton, H.: Choosing the right lens: Finding what is interesting in data mining. In: Guillet, F., Hamilton, H. (eds.) Quality Measures in Data Mining, pp. 3–24. Springer, Heidelberg (2007)
7. Lallich, S., Teytaud, O., Prudhomme, E.: Association rule interestingness: Measure and statistical validation. In: Guillet, F., Hamilton, H. (eds.) Quality Measures in Data Mining, pp. 251–275. Springer, Heidelberg (2007)
8. Bay, S., Pazzani, M.: Detecting group differences. mining contrast sets. Data Mining and Knowledge Discovery 5, 213–246 (2001)
9. Hilderman, R., Peckham, T.: Statistical methodologies for mining potentially interesting contrast sets. In: Guillet, F., Hamilton, H. (eds.) Quality Measures in Data Mining, pp. 153–177. Springer, Heidelberg (2007)
10. Lerman, L., Azè, J.: Une mesure probabiliste contextuelle discriminante de qualite des règles d'association. In: EGC 2003: Extraction et Gestion des Connaissances, pp. 247–263. Hermes Science Publications, Lavoisier (2003)
11. Lenca, P., Vaiilant, B., Meyer, P., Lalich, S.: Association rule interestingness meaures: Experimental and theoretical studies. In: Guillet, F., Hamilton, H. (eds.) Quality Measures in Data Mining, pp. 51–76. Springer, Heidelberg (2007)
12. Hájek, P., Havránek, T.: Mechanizing Hypothesis Formation. Springer, Berlin (1978)
13. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.: Fast discovery of association rules. In: Advances in Knowledge Discovery and Data Mining, pp. 307–328. AAAI Press, Menlo Park (1996)

14. Zaki, M., Parathasarathy, S., Ogihara, M., Li, W.: New parallel algorithms for fast discovery of association rules. Data Mining and Knowledge Discovery 1, 343–373 (1997)
15. Korn, F., Labrinidis, A., Kotidis, Y., Faloutsos, C.: Quantifiable data mining using ration rules. VLDB Journal 8, 254–266 (2000)
16. Michalski, R.: Knowledge acquisition through conceptual clustering: A theoretical framework and algorithm for partitioning data into conjunctive concepts. International Journal of Policy Analysis and Information Systems 4, 219–243 (1980)
17. Michalski, R., Kaufman, K.: Learning patterns in noisy data. In: Machine Learning and Its Applications, pp. 22–38. Springer, New York (2001)
18. Clark, P., Boswell, R.: Rule induction with CN2: Some recent improvements. In: Kodratoff, Y. (ed.) Machine Learning - EWSL-91. LNCS, vol. 482, pp. 151–163. Springer, Heidelberg (1991)
19. De Raedt, L.: Interactive Theory Revision: An Inductive Logic Programming Approach. Academic Press, London (1992)
20. Muggleton, S.: Inductive Logic Programming. Academic Press, London (1992)
21. Freitas, A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer, Berlin (2002)
22. Jang, J.: ANFIS: Adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man, and Cybernetics 23, 665–685 (1993)
23. Jang, J., Sun, C.: Neuro-fuzzy modeling and control. The Proceedings of the IEEE 83, 378–406 (1995)
24. Nauck, D., Kruse, R.: NEFCLASS-X: A neuro-fuzzy tool to build readable fuzzy classifiers. BT Technology Journal 3, 180–192 (1998)
25. Nauck, D.: Fuzzy data analysis with NEFCLASS. International Journal of Approximate Reasoning 32, 103–130 (2002)
26. Holeňa, M.: Fuzzy hypotheses for Guha implications. Fuzzy Sets and Systems 98, 101–125 (1998)
27. Holeňa, M.: Fuzzy hypotheses testing in the framework of fuzzy logic. Fuzzy Sets and Systems 145, 229–252 (2004)
28. Novák, V., Perfilieva, I., Dvořák, A., Chen, C., Wei, Q., Yan, P.: Mining pure linguistic associations from numerical data. International Journal of Approximate Reasoning (to appear)
29. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth, Belmont (1984)
30. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Francisco (1992)
31. Finn, G.: Learning fuzzy rules from data. Neural Computing & Applications 8, 9–24 (1999)
32. Duch, W., Adamczak, R., Grabczewski, K.: A new methodology of extraction, optimization and application of crisp and fuzzy logical rules. IEEE Transactions on Neural Networks 11, 277–306 (2000)
33. Tsukimoto, H.: Extracting rules from trained neural networks. IEEE Transactions on Neural Networks 11, 333–389 (2000)
34. Holeňa, M.: Piecewise-linear neural networks and their relationship to rule extraction from data. Neural Computation 18, 2813–2853 (2006)
35. Tickle, A., Andrews, R., Golea, M., Diederich, J.: The truth will come to light: Directions and challenges in extracting rules from trained artificial neural networks. IEEE Transactions on Neural Networks 9, 1057–1068 (1998)
36. Mitra, S., Hayashi, Y.: Neuro-fuzzy rule generation: Survey in soft computing framework. IEEE Transactions on Neural Networks 11, 748–768 (2000)

37. Šimunek, M.: Academic KDD project LISp-miner. In: Abraham, A., Franke, K., Koppen, K. (eds.) Advances in Soft Computing – Systems Desing and Applications, pp. 263–272. Springer, Heidelberg (2003)
38. Hájek, P., Holeňa, M.: Formal logics of discovery and hypothesis formation by machine. Theoretical Computer Science 292, 345–357 (2003)
39. Holeňa, M.: Extraction of fuzzy logic rules from data by means of artificial neural networks. Kybernetika 41, 297–314 (2005)
40. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer Academic Publishers, Dordrecht (1998)
41. Holeňa, M.: Neural networks for extraction of fuzzy logic rules with application to EEG data. In: Ribeiro, B., Albrecht, R., Dobnikar, A. (eds.) Adaptive and Natural Computing Algorithms, pp. 369–372. Springer, Heidelberg (2005)
42. Fisher, R.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179–188 (1936)

A Discriminative Learning Method of TAN Classifier

Qi Feng, Fengzhan Tian, and Houkuan Huang

School of Computer & Information Technology, Beijing Jiaotong University,
100044, Beijing, China
fengqi2008@gmail.com

Abstract. TAN (Tree-augmented Naïve Bayes) classifier makes a compromise between the model complexity and classification rate, the study of which has now become a hot research issue. In this paper, we propose a discriminative method that is based on KL (Kullback-Leibler) divergence to learn TAN classifier. First, we use EAR (explaining away residual) method to learn the structure of TAN, and then optimize TAN parameters by an objective function based on KL divergence. The results of the experiments on benchmark datasets show that our approach produces better classification rate.

Keywords: TAN classifier, discriminative learning, KL divergence, EAR.

1 Introduction

Classification is an important research issue in the fields of Machine Learning, Data Mining and Pattern Recognition. Bayesian classifier is a powerful classification model which is on the basis of Bayes statistics and Bayesian Networks. In particular, Naïve Bayes classifier [1] is the simplest, effective and widely used Bayesian classifier, but it assumes that all the attributes are conditionally independent given the class label. In order to improve the performance of Naïve Bayes classifier, Nir Friedman et al. [2] introduced the TAN (Tree-augmented Naïve Bayes) classifier which released the conditional independence restriction. They showed that in the structure of TAN classifier, each attribute has the class node and at most one other attribute as its parents. TAN classifier is more accurate than Naïve Bayes classifier on most of benchmark datasets.

GBN (General Bayesian Network) classifier is another kind of classifier which further generalizes the conditional independence assumption. Because there is no restriction to the relationship between each pair of attributes, GBN is the most accurate network that represents realistic problem and would achieve a higher classification rate theoretically. But Friedman et al. found that Naïve Bayes classifier outperforms GBN classifier on a large number of benchmark datasets. The reason is that the objective function used in the standard learning of Bayesian classifier attempts to optimize the joint probability of attributes and the class variable. Then perform classification by computing the posterior probability of class variable with Bayes rule. This method is known as generative learning. Because of the mismatch between the objective function and the goal of classification, generative learning of

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 443–452, 2007.

Bayesian classifiers could not achieve the highest classification accuracy though it can accurately approximate the entire data.

So how to construct a Bayesian classifier with higher classification accuracy becomes a hot research issue in recent years. The discriminative learning of Bayesian classifier attempts to learn a discriminant function or model the class posterior probability directly. A standard approach to learn discriminative classifiers directly optimizes the conditional likelihood of class variable given the attributes in the datasets. The coherence of the objective function and classification task makes discriminative classifier achieve higher accuracy.

Discriminative learning of Bayesian classifiers has attracted widespread attention in recent years, most approaches of which trained the structures or the parameters of Bayesian classifiers by maximizing the conditional likelihood. As for fixed structures, Greiner and Zhou [3] introduced an optimization algorithm named ELR for the parameter learning. This approach computes the maximum conditional likelihood parameters by a conjugate gradient method. ELR can learn the parameters of any arbitrary structure and works better than generative parameter learning methods. Grossman et al. [4] used conditional likelihood to learn the structures of the Bayesian networks while the parameters are set by "observed frequency estimate" (OFE)[5]. This approach use a hill-climbing search: at each search step, add, delete, or reverse a single arc in the structure and score the model until the conditional likelihood score gets a maximum. Pernkopf et al. [6] trained the Bayesian classifiers by learning discriminatively both parameters and structures.

In this paper, we introduce a discriminative learning method of TAN classifier based on KL (Kullback-Leibler) divergence. First, we use EAR (explaining away residual) method to learn the structure of TAN classifier, and then optimize the parameters of TAN classifier with an objective function based on KL divergence. Experimental results on benchmark datasets from UCI repository show that our approach to learning TAN classifier produces better classification accuracy than generative learning methods.

This paper is organized as follows: In section 2, we introduce some basic concepts of Bayesian classifier. In section 3, we present the discriminative approach to learning both the TAN structure and the TAN parameters. In section 4, we report the experimental results on benchmark datasets from UCI repository. And we draw conclusions in section 5.

2 Bayesian Network Classifier

Given an i.i.d (independent identically distributed) training dataset $D = \{Z_1, \ldots, Z_d, \ldots, Z_N\}$, where Z_d represents an instance. Each instance Z is denoted by a lower case vector $\{x_1, x_2, \ldots, x_n, x_{n+1}\}$, where the first n random variables denote the special values taken by n attributes $X_1, X_2, \ldots, X_n$ of the Bayesian classifier and the random variable x_{n+1} ($x_{n+1} \in C = \{c_1, c_2, \ldots, c_m\}$) denotes the value of class label X_{n+1}, and m is the cardinality of the set C.

Given an unknown instance in which the class label is unsigned, the goal of classification is to correctly predict the value of class variable.

A Bayesian classifier perform classification task by calculating the posterior probability of class label given predictive attributes using Bayes rule [7]:

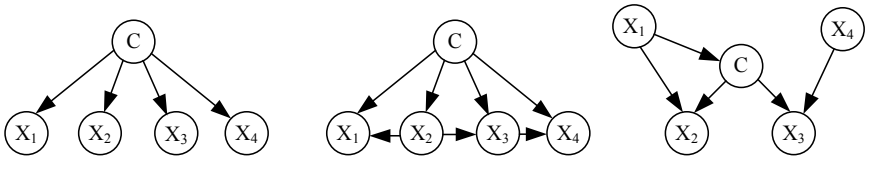
$$P(c_i \mid x_1,...,x_n,\theta) = \frac{P(x_1,...,x_n,c_i \mid \theta)}{\sum_{c'=1}^{m} P(x_1,...,x_n,c' \mid \theta)} \cdot \tag{1}$$

Where θ is the parameters of the Bayesian classifier.

Generally speaking, a Bayesian classifier takes all the possible assumptions such as c_i $(i = 1, 2, \ldots, m)$ into account and chooses the maximal posterior (maximum a posteriori, MAP)[8] as the class value. So a vector of attributes $\{x_1, x_2, \ldots, x_n\}$ is designated as c_i if and only if

$$c_i = \arg\max_{1 \le j \le m} \{P(c_j \mid x_1,...,x_n)\} \cdot \tag{2}$$

Naïve Bayes classifier has a simple structure as shown in fig. 1 (a), in which each attribute has the class label as its parent and the class has no parent. It captures the main assumption that each attribute is independent from other attributes given the state of the class variable. But this assumption is unrealistic. In order to correct the limitation of Naïve Bayes classifier, Friedman et al. [2] presented TAN classifier. A TAN is a structural augmentation of Naïve Bayes classifier, where the class variable has no parents and every attribute has the class variable and at most one other attribute as its parents. The structure of TAN classifier is shown in fig. 1 (b). GBN (General Bayesian Network) is a more complex model of Bayesian classifiers. GBN deals with the class variable as a general attribute, and it represents the factual relationship between all the variables. The structure of GBN is illustrated as fig. 1 (c).



(a) Naïve Bayes classifier strcuture (b) TAN classifier strcuture (c) GBN classifier strcuture

Fig. 1. Structures of various Bayesian network classifiers

Learning a Bayesian network classifier corresponds to the fact to find a model that veritably describes the relationship between the variables according to the given data. There are two steps while learning a TAN classifier. The first step is to determine a structure of TAN. In order to construct a directed acyclic graph, we should figure out the parents for each variable. The second step is to specify the parameters given the TAN structure. That is to calculate the conditional probability distribution of the TAN structure.

Chow and Liu described an algorithm [9] to construct a tree-like Bayesian network. Friedman et al. introduced a procedure [2] to learn the structure of TAN classifier. This method follows the general outline of Chow and Liu's procedure, except that it uses conditional mutual information (CMI) between two attributes given

the class label instead of mutual information between the two attributes. The procedure is illustrated as follows:

a. Compute the CMI between each pair of attributes X_i, X_j ($i \neq j$).

$$I(X_i, X_j \mid C) = \sum_{x_i, x_j, c} P(x_i, x_j, c) \cdot \log \frac{P(x_i, x_j \mid c)}{P(x_i \mid c) P(x_j \mid c)} \cdot \tag{3}$$

b. Build a complete undirected graph in which the vertices are the attributes X_1, X_2, …, X_n. Note that the weight of each edge connecting X_i and X_j is the CMI value between them.

c. Build a maximum weighted spanning tree.

d. Choose a variable as the root node and set the direction of each edge to be outward from it in order to transform the undirected graph into a directed graph.

e. Construct a TAN model by adding class vertex C and adding the arc from C to each X_i.

As each attribute variable has the class variable and at most one other attribute as its parents, we denote the parameters of TAN as the following form:

$$\theta_{ijk} = P_c(X_i = k \mid \pi_i = j) \ . \tag{4}$$

Where c represents the value of class variable and π_i represents another parent of vertex X_i except for the class variable. Note that if vertex X_i has only one parent, the class variable, π_i is null. This parameter denotes a special conditional probability table entry that the variable X_i takes the kth value assignment given that its parent π_i takes the jth assignment.

Generative learning method optimizes the parameters of a Bayesian classifier by maximizing the likelihood (ML) of data. Given a directed acyclic graph S of a Bayesian network classifier B, the log likelihood objective function is

$$LL(B \mid S) = \sum_{d=1}^{N} \log P_\theta(Z^d) = \sum_{d=1}^{N} \sum_{i=1}^{n+1} \log P_c(x_i^d \mid \pi_i^d) \cdot \tag{5}$$

In this function, Z^d is the dth instance of training dataset denoted by a lower case vector $\{x_1{}^d, x_2{}^d, \ldots, x_n{}^d, x_{n+1}{}^d\}$ and N is the total number of instances in the dataset. This joint likelihood can be decomposed as a linear form of the log of parameters. Each parameter for the instance is denoted as $P_c(X_i^d = k \mid \pi_i^d = j)$.

It is easy to calculate the parameters of ML estimate which are just observed frequency estimates. After the generative learning procedures of TAN structure and parameters, we have already learned a TAN classifier and could perform classification with it.

3 Discriminative Learning of TAN Classifier Based on KL Divergence

In this paper, we learn the TAN classifier in a discriminative way. First we use the procedure proposed by Friedman to learn the TAN structure, in which we compute

the EAR (explaining away residual) value instead of CMI between each pair of variables. Given the TAN structure, we then estimate the parameters by optimizing an objective function based on KL divergence.

3.1 Discriminative learning of TAN structure

It is a generative approach to learn the structure of a TAN classifier by computing CMI, since CMI produces a guaranteed nondecrease in the joint likelihood over all variables while augmenting the structure [10].

Bilmes introduced a discriminative criterion named explaining away residual (EAR)[10] which calculates the subtraction between CMI and unconditional mutual information.

$$I(X_i, X_j \mid C) - I(X_i, X_j) = \sum_{x_i, x_j, c} P(x_i, x_j, c) \cdot \log \frac{P(x_i, x_j \mid c)}{P(x_i \mid c) P(x_j \mid c)} - \sum_{x_i, x_j} P(x_i, x_j) \cdot \log \frac{P(x_i, x_j)}{P(x_i) P(x_j)} \quad . \tag{6}$$

Bilmes [11] has proved that optimizing the EAR value is equivalent to decreasing the divergence between the true posterior and the resultant approximate posterior. Improving EAR measure is in fact an approximation to maximizing the log conditional likelihood. So EAR method is a discriminative approach of TAN structure learning.

In this paper, we use EAR measure to construct TAN structure. But we add an edge between two variables if necessary when their EAR value is larger than zero, as we are interested in the extra information that X_j provides to X_i given the class variable.

3.2 Discriminative Parameter Learning Based on KL Divergence

In information theory, KL (Kullback-Leibler) divergence [12] is used to depict the difference between two distributions. And it is also called cross entropy. Given two distribution $p(x)$ and $q(x)$, the KL divergence between $p(x)$ and $q(x)$ is

$$KL(p;q) = \sum_x p(x) \cdot \log \frac{p(x)}{q(x)} \quad . \tag{7}$$

In Bayesian learning, we use KL divergence to describe the difference between the empirical distribution $\hat{p}$ and the true distribution p. Kaizhu Huang optimizes an objective function [13] including KL divergence to solve the parameter learning problem of Naïve Bayes classifier, and the experimental results show that discriminative Naïve Bayes classifier outperforms the generative one. In this paper, we introduce KL divergence into discriminative parameter learning of TAN classifier by optimizing a discriminative objective function similar to that in SVM [14].

In a two-class classification problem, we can partition a dataset into two subsets S_1 and S_2 according to the value of class variable. We use $\hat{p}_1$ and $\hat{p}_2$ to represent respectively the empirical distributions for subset S_1 and S_2, which are initialized as

the observed frequency estimates. We use p_1 and p_2 to represent the corresponding true distributions.

For discriminative Bayesian learning, we should not only approximate the distribution of each subset as accurately as possible, but also augment the divergence between true distributions of the two subsets. So we can use the following formula as a measure function:

$$\min\{KL(\hat{p}_1, p_1) + KL(\hat{p}_2, p_2) + W \cdot Div(p_1, p_2)\} . \tag{8}$$

In the objective function, the first two items describe how accurate the distribution p_1 and p_2 approximate the two sub-datasets S_1 and S_2, and they provide the inner-class information. The last item is the key part in discriminative learning, which describes the inter-class information. Namely, this item makes the divergence between two distributions of the two classes as big as possible. W is a penalty factor. The larger this factor, the bigger divergence between classes. If this factor is 0, the objective function contains only the first two items which make the TAN parameters approximate the dataset. We use the opposite of KL divergence measure to represent the third item.

$$Div(p_1, p_2) = -KL(p_1, p_2) . \tag{9}$$

The objective function can be written as follows:

$$\min \left\{ \begin{array}{l} \sum_{c=1}^{2} \sum_{i=1}^{n} \sum_{j=1}^{|\pi_i|} \sum_{k=1}^{|x_i|} \hat{p}_c(X_i = k \mid \pi_i = j) \cdot \log \frac{\hat{p}_c(X_i = k \mid \pi_i = j)}{p_c(X_i = k \mid \pi_i = j)} \\ -W \cdot \sum_{i=1}^{n} \sum_{j=1}^{|\pi_i|} \sum_{k=1}^{|x_i|} p_1(X_i = k \mid \pi_i = j) \cdot \log \frac{p_1(X_i = k \mid \pi_i = j)}{p_2(X_i = k \mid \pi_i = j)} \end{array} \right\} \tag{10}$$

$$\text{s.t.} \quad 0 \le p_c(X_i = k \mid \pi_i = j) \le 1 ,$$

$$\sum_{k=1}^{|x_i|} p_c(X_i = k \mid \pi_i = j) = 1 , \text{ where } c = 1, 2; i = 1, \ldots, n .$$

Where c represents the value of the class variable, and for the two-class classification, c takes value of 1 and 2. $\hat{p}_c(X_i = k \mid \pi_i = j)(c=1, 2; i=1, 2, \ldots, n)$ is the empirical distribution with respect to the dataset and $p_c(X_i = k \mid \pi_i = j)(c=1, 2, i=1, 2,\ldots, n)$ is the true distribution that is to be optimized.

So far we have transformed the discriminative parameter learning problem into a nonlinear optimization problem under linear constraints. This optimization problem can be solved by mathematic approach such as gradient projection method.

4 Experimental Analysis

To evaluate the performance of our discriminative TAN classifier, we did experiments on 10 benchmark datasets from Machine Learning Repository in UCI. The detailed information of these datasets is listed in Table 1.

Table 1. Description of datasets used in experiments

Dataset name	# Attribute	# Class	# Instance
AUSTRALIAN	14	2	690
CLEVELAND	13	2	303
DIABETES	8	2	768
DMPLEXER	14	2	1000
GERMAN	20	2	1000
HEART	13	2	270
HORSE-COLIC	22	2	368
HUNGARIAN	13	2	294
SONAR	60	2	208
TIC-TAC-TOE	9	2	958

The attributes in the datasets are discrete nominal or continuous numeric, but TAN classifier can only deal with nominal attributes. All the continuous attributes are discretized before used for classification. We use Laplace estimation instead of the zero probability in the conditional probability tables. Namely, a conditional probability is adjusted with Laplace correction values, 1 is added to the numerator and the number of values of the attribute is added to the denominator.

All the experiments are done based on Weka system [15]. The TAN classifiers are trained with generative and discriminative parameters on generative and discriminative structures. And we compare our discriminative TAN classifier with other 3 classifiers. In our experiments, we use CMI as a generative approach, EAR as a discriminative method for structure learning of TAN. The parameter training is performed with ML and KL, where ML is a generative way and KL is utilized as a discriminative method. The penalty factor W is set as 0.05. We use 5-fold cross-validation to evaluate the performance of these classifiers.

Table 2 shows the experimental results of all the four classifiers over the datasets in Table 1. The bottom line in Table 2 gives the average classification accuracy of each classifier over the ten datasets. From Table 2, we find that CMI+KL classifier outperforms CMI+ML on 5 of 10 datasets, and losses on 2 datasets. EAR+KL works better than EAR+ML on 4 datasets, and losses on 2 datasets. On the other hand, EAR+ML classifier outperforms CMI+ML and EAR+KL exceeds CMI+KL in terms of classification accuracy. Finally, we see that the EAR+KL TAN classifier achieves the highest classification rate on 6 datasets and obtains the best performance on average. This suggests that the discriminative parameter learning method based on KL divergence and discriminative structure learning method are always effective to improve classification accuracy compared with generative methods.

In the above experiments, penalty factor W in objective function is set empirically as 0.05. Because the choice of this factor would influence the classification result, we have done experimental study on parameter sensitivity before our experiments and list our analyses on the following two datasets.

Table 2. Experiment results: Classification accuracy (%) of 4 TAN classifiers

Dataset name	TAN-CMI-ML	TAN-CMI-KL	TAN-EAR-ML	TAN-EAR-KL
AUSTRALIAN	*84.9275*	83.6232	*84.9275*	83.4783
CLEVELAND	81.1881	81.1881	*82.5083*	*82.5083*
DIABETES	*77.6042*	76.5625	77.2135	76.4323
DMPLEXER	54.5	55.1	56.1	*56.7*
GERMAN	74.4	74.4	*75.1*	*75.1*
HEART	80.7407	82.2222	81.8519	*82.5926*
HORSE-COLIC	81.25	*81.7935*	81.25	81.25
HUNGARIAN	81.6327	81.6327	*83.3333*	*83.3333*
SONAR	75.4808	76.4423	76.9231	*77.4038*
TIC-TAC-TOE	76.5136	*77.7662*	71.6075	72.2338
Average	76.8238	77.0731	77.0815	*77.1032*

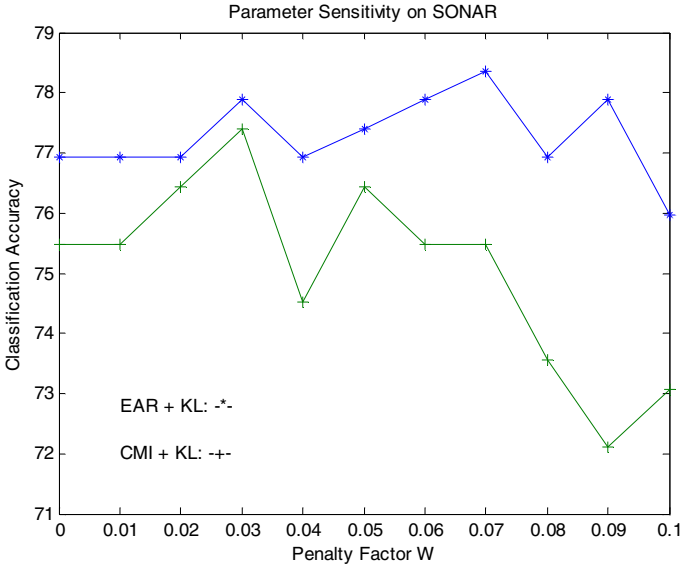


Fig. 2. Classification Accuracy on dataset SONAR

Fig. 2 and Fig. 3 show the sensitivity of the penalty factor *W*. The star solid line represents classification accuracy of EAR+KL method on the datasets, and plus solid line denotes that of CMI+KL algorithm. In these accuracy curves, the horizontal axis lists the values of the penalty factor *W* which is ranging from 0 to 0.1, and the vertical axis measures the classification accuracy.

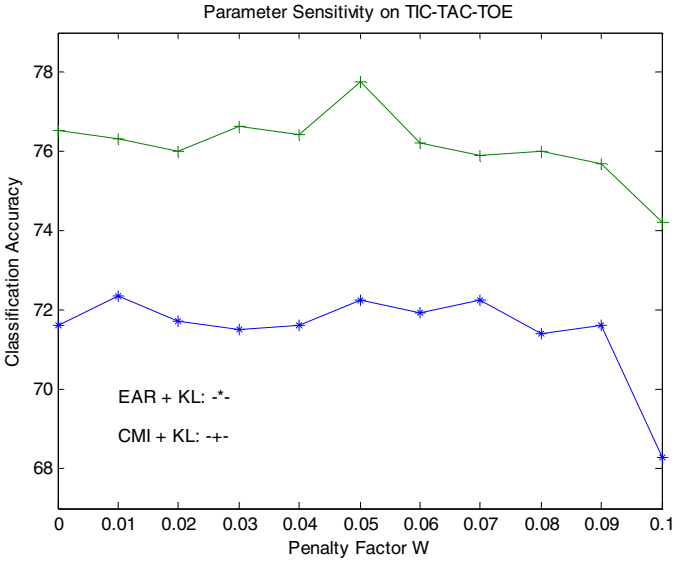


Fig. 3. Classification Accuracy on dataset TIC-TAC-TOE

We can empirically find that when penalty factor is set as 0, the classification results of EAR+KL and CMI+KL methods are the same as those of EAR+ML and CMI+ML. In Fig. 2, EAR+KL model reaches a maximum of 78.3654% when the penalty factor is 0.07, but CMI+KL model reaches a maximum 77.4038% at 0.03. We can make a compromise when the penalty factor is 0.05. In Fig. 3, both EAR+KL and CMI+KL models reach the most accurate classification results at 0.05. The bigger the penalty factor value, the worse the accuracy. So we set the penalty factor W as 0.05 in all our experiments.

5 Conclusion and Future Work

This paper proposes a discriminative learning method of TAN classifier based on KL divergence. We use EAR (explaining away residual) method for discriminative learning of TAN structure and optimize TAN parameters with an objective function based on KL divergence. The experimental results show that our approach to learning TAN classifier could achieve better accuracy than others.

In this paper, we present our discriminative TAN learning algorithm for two-class classification problems but not multi-class ones. So a direction of our future work includes extending our algorithm to handle multi-class classification problem. Our future work also includes applying our discriminative learning method to a wider variety of datasets and choosing the penalty factor W in objective function more automatically according to the datasets.

Acknowledgments

This work is supported by NSF of China under Grant No. 60503017, Beijing Nova Program under Grant No. 2006A17 and Science Foundation of Beijing Jiaotong University under Grant No. 2005SM012. Thank anonymous readers for helpful comments.

References

1. Duda, R., Hart, P.: Pattern Classification and Scene Analysis. John Wiley &Sons, New York (1973)
2. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian Network Classifiers. Machine Learning 29, 131–163 (1997)
3. Greiner, R., Zhou, W.: Structural Extension to Logistic Regression: Discriminative Parameter Learning of Belief Net Classifiers. In: 18th Conf. of the AAAI, pp. 167–173 (2002)
4. Grossman, D., Domingos, P.: Learning Bayesian Network Classifiers by Maximizing Conditional Likelihood. In: 21st Inter. Conf. of Machine Learning (ICML), pp. 361–368 (2004)
5. Cooper, G., Herskovits, E.: A Bayesian Method for the Induction of Probabilistic Networks from Data. Machine Learning 9, 309–347 (1992)
6. Pernkopf, F., Bilmes, J.: Discriminative versus Generative Parameter and Structure Learning of Bayesian Network Classifiers. In: 22nd Inter. Conf. of Machine Learning (ICML) (2005)
7. Han, J.W., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, Inc, San Francisco (2001)
8. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
9. Chow, C.K., Liu, C.N.: Approximating Discrete Probability Distributions with Dependence Trees. IEEE Trans. on Info. Theory 14, 462–467 (1968)
10. Bilmes, J.: Dynamic Bayesian multinets. In: 16th Inter. Conf. of Uncertainty in Artificial Intelligence (UAI), pp. 38–45 (2000)
11. Bilmes, J.: Natural Statistical Models for Automatic Speech Recognition. Doctoral dissertation, U.C. Berkeley (1999)
12. Cover, T.M., Thomas, J.A.: Elements of information theory. Wiley & Sons, New York (1991)
13. Huang, K.Z.: Discriminative Naïve Bayesian Classifiers (2003), http://citeseer.ist.psu.edu/600619.html
14. Vapnik, V.N.: The Nature of statistical Learning Theory. Springer, New York (1995)
15. Witten, I.H., Frank, E.: Data Ming: Practical Machine Learning Tools and Techniques with Java Implementations, pp. 265–314. Morgan Kaufimann Publishers, Seattle (2000)

Discriminative vs. Generative Learning of Bayesian Network Classifiers

Guzmán Santafé, Jose A. Lozano, and Pedro Larrañaga

Intelligent Systems Group
Department of Computer Science and Artificial Intelligence
University of the Basque Country, Spain
guzman@si.enu.es, {ja.lozano, pedro.larranaga}@ehu.es

Abstract. Discriminative learning of Bayesian network classifiers has recently received considerable attention from the machine learning community. This interest has yielded several publications where new methods for the discriminative learning of both structure and parameters have been proposed. In this paper we present an empirical study used to illustrate how discriminative learning performs with respect to generative learning using simple Bayesian network classifiers such as naive Bayes or TAN, and we discuss when and why a discriminative learning is preferred. We also analyzed how log-likelihood and conditional log-likelihood scores guide the learning process of Bayesian network classifiers.

1 Introduction

Supervised classification is a part of machine learning used in many fields such as bioinformatics, computer vision, speech recognition or medical diagnosis. In general, supervised classification problems are defined in terms of two different kinds of variables: the predictive variables, $\boldsymbol{X} = (X_1, \ldots, X_n)$, and the class variable or response, C. A supervised classifier attempts to learn the relation between the predictive and the class variables. Hence, it is able to assign a class value to a new data sample $\boldsymbol{x} = (x_1, \ldots, x_n)$ whose response is unknown.

Over the last few years, Bayesian networks [1,2] have received considerable attention from the machine learning community since they are powerful probabilistic tools that enable a simple and efficient representation of probability distributions. Furthermore, there is an outstanding use of Bayesian networks as classifiers in supervised classification problems [3].

A Bayesian network is defined as a pair $\mathcal{B} = \{\mathcal{G}, \boldsymbol{\Theta}\}$, where $\mathcal{G}$ is a directed acyclic graph whose arcs define the (in)dependencies between the variables and $\boldsymbol{\Theta}$ are the parameters of the model that represent the conditional probability tables for the factorization given by the graph. Thus, given a Bayesian network model $\mathcal{B}$, we can write the joint probability distribution as:

$$p(\boldsymbol{x}|\mathcal{B}) = \prod_{i=0}^{n} p(x_i|\boldsymbol{pa}_i, \mathcal{B}) \tag{1}$$

where X_0 denotes, in this case, the class variable C, and $\boldsymbol{pa}_i$, with $i = 0, ..., n$ represents the value of the parents of X_i, being the parents of X_i ($\boldsymbol{Pa}_i$) defined by $\mathcal{G}$.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 453–464, 2007.

In general, learning a Bayesian network classifier requires finding a network structure (structural learning) that represents the (in)dependencies among the variables in the problem, and then determining the appropriate parameters for that structure (parametric learning). The structural learning searches through the space of directed acyclic graphs for the one that observes the model restrictions and optimizes a model selection criterion. Thus, by setting model restrictions we reduce the model complexity. Naive Bayes and tree augmented naive Bayes (TAN) are probably the most popular Bayesian network classifiers. Naive Bayes assumes that all the predictive variables are independent given the class variable [4], while TAN [5] relaxes naive Bayes assumption by allowing the predictive variables to form up to a tree. Additionally, in the literature there are other proposals that set different restrictions in the model structure. For instance, k-dependent Bayesian classifier [6] allows predictive variables to have up to k predictive variables as parents, and Bayes network augmented naive Bayes (BAN) [7] makes no restrictions in the relations between predictive variables.

Both structural and parametric methods to learn a Bayesian network classifier are often described as either generative or discriminative. A generative method to learn Bayesian network classifiers models the joint probability distribution of the predictive variables and the class, $p(C, \boldsymbol{X})$, while a discriminative method models the conditional probability distribution of the class given the predictive variables, $p(C|\boldsymbol{X})$. Bayesian network classifiers are considered generative models since the learning process usually maximizes the log-likelihood (LL) which is defined as the probability of the dataset D given the model $\mathcal{B}$:

$$LL = \sum_{l=1}^{N} \log p(c^{(l)}, \boldsymbol{x}^{(l)}|\mathcal{B}) \tag{2}$$

where N is the number of samples in dataset D. Additionally, we abuse the notation by writing $c^{(l)}$ and $\boldsymbol{x}^{(l)}$ to represent the fact that C and $\boldsymbol{X}$ take the values given in the l-th sample of the dataset D.

On the other hand, there have been a number of recent proposals in the literature for a discriminative learning of both structure and parameter of Bayesian networks classifiers by maximizing the conditional log-likelihood (CLL):

$$CLL = \sum_{l=1}^{N} \log p(c^{(l)}|\boldsymbol{x}^{(l)}, \mathcal{B}) \tag{3}$$

Greiner et al. [8] introduce a conjugate gradient method to obtain the parameters of the model by maximizing CLL. Similarly, Roos et al. [9] and Feelders and Ivanovs [10] propose ways to map the parameters of Bayesian network classifiers to logistic regression models in order to obtain the parameters of the Bayesian network classifiers that maximizes CLL. Santafé et al. [11] present an algorithm based on sufficient statistics which is able to obtain the parameters that maximize the CLL score for Bayesian network classifiers. Moreover, there are a few proposals for the discriminative learning of the structure: Grossman and Domingos [12] use CLL and a hill climbing method to learn the structure of a Baysian network classifier but the parameters are learned with a generative method;

Santafé et al. [13] learn both structure and parameter by maximizing CLL; Guo and Greiner [14] and Pernkopf and Bilmes [15] compare the performance of different discriminative approaches to learn Bayesian network classifiers.

It is thought [16,17] that discriminative learning should be preferred for classification purposes since it directly models the discriminative function $p(C|\boldsymbol{X})$. A generative approach, on the contrary, deals with a more general problem, modeling $p(C, \boldsymbol{X})$, to solve a more specific one, model $p(C|\boldsymbol{X})$. However, in practice, discriminative approaches do not always perform better than generative approaches [18].

In this work, we present a comparison between generative and discriminative learning for structure and parameters of naive Bayes and TAN models. We perform experiments that attempt to illustrate some hypotheses about discriminative and generative learning that have been stated in the literature. Additionally, we also evaluate how LL and CLL scores are able to guide the learning of both structure and parameters.

The rest of the paper is organized as follows. In Section 2 we introduce the Bayesian network classifiers used in the paper as well as the generative and discriminative methods used to learn the structure and the parameters of these classifiers. Section 3 presents an empirical study of generative and discriminative learning methods using synthetic and real datasets. These experiments are used to illustrate the behavior of both generative and discriminative methods under different conditions. Finally, Section 4 exposes some conclusions yielded from the paper.

2 Classification Models

In this paper, we take into consideration naive Bayes and TAN models. Naive Bayes does not require the learning of the structure since it does not depend on the dataset but only on the number of variables and their states. However, its parameters can be learned by either generative or discriminative methods. On the other hand, TAN models need for both structural and parametric learning which can be also approached by either generative or discriminative methods.

2.1 Generative Learning of TAN Structures

In order to learn a generative TAN structure, we use the original method proposed by Friedman et al. [5]. It maximizes the LL score by constructing a maximum spanning tree using the mutual information between each pair of predictive variables given the class variable.

$$I(X_i, X_j|C) = \sum_{x_i,x_j,c} p(x_i, x_j, c) \log \frac{p(x_i, x_j|c)}{p(x_i|c)p(x_j|c)} \tag{4}$$

with $i = 1, \ldots, n$; $j = 1, \ldots, n$ and $i \neq j$.

2.2 Discriminative Learning of TAN Structures

Conditional log-likelihood can be maximized when learning TAN structures by maximizing a metric known as *explaining away residual* (EAR) (see Equation 5). This score was introduced by Bilmes [19] and also used in [15] and [20] to learn discriminative TAN models.

$$EAR = I(X_i, X_j|C) - I(X_i, X_j) \tag{5}$$

The EAR is in fact an approximation to the expected log-posterior. Therefore, the maximization of EAR is equivalent to maximize CLL.

The learning algorithm used to learn a discriminative structure of a TAN is similar to Friedman et al.'s algorithm [5] but using the EAR metric instead of $I(X_i, X_j|C)$ in order to guide the structural search [15,20]. In this paper, we name the TAN model whose structure is learned by the method described above as conditional TAN (cTAN) model.

2.3 Generative Learning of Parameters

A generative learning of the parameters of a model can be performed by maximizing the LL score, that is, the well-known maximum likelihood estimation of the parameters. This is the most used method because the decomposability of the LL score leads to a simple and efficient estimation of the parameters. The maximum likelihood parameters are given by:

$$\theta_{ijk} = \frac{N_{ijk}}{N_{ij}} \tag{6}$$

where $\theta_{ijk} \in \boldsymbol{\Theta}$ represents the conditional probability $p(x_i^k|\boldsymbol{pa}_i^j)$, N_{ijk} is a sufficient statistic of the dataset that denotes the number of samples where variable X_i takes its k-th (x_i^k) value and $\boldsymbol{Pa}_i$ their j-th configuration ($\boldsymbol{pa}_i^j$). Additionally, $N_{ij} = \sum_k N_{ijk}$.

2.4 Discriminative Learning of Parameters

In order to maximize CLL when learning the parameters of a model, we use the TM algorithm proposed by Santafé et al. [11]. This is a variation of the general TM algorithm [21] adapted to the discriminative learning of the parameters in Bayesian network classifiers. The general idea of the algorithm is that the CLL can be approximated, using an iterative process, by a function based only on the LL. Thus, we are able to map the sufficient statistics of the conditional model, $p(C|\boldsymbol{X})$, with those in the unconditional model, $p(C, \boldsymbol{X})$, and therefore, it is possible to obtain the parameters that maximize the CLL by maximizing the approximation of CLL in terms of LL.

3 Discriminative vs. Generative Learning

There is a strong belief in the scientific community that discriminative learning has to be preferred in reasoning tasks. The maximization of the LL score does

not necessarily lead to improve the classification rate [5]. In fact, LL score not only depends on CLL but also on the marginal log-likelihood:

$$LL = \sum_{l=1}^{N} \log p(c^{(l)}|\boldsymbol{x}^{(l)}), \mathcal{B}) + \sum_{l=1}^{N} \log p(\boldsymbol{x}^{(l)}|\mathcal{B}) \tag{7}$$

As can be seen in Equation 7, only the first term (CLL) is related to classification since the second one, marginal likelihood, is only relevant to model the relation between predictive variables but not for classification purposes. Moreover, as the number of predictive variables increases, the second term in Equation 7 should become much more relevant than the first one. Therefore, a criteria related to the LL score might be adequate to learn the (in)dependencies between variables captured in the dataset but could be inadequate to learn a classification model, especially with a high number of predictive variables. By contrast, the CLL score is directly related to the discrimination function $p(c|\boldsymbol{x})$. The maximization of CLL also involves minimizing the entropy of the class given the predictive variables, $H(C|\boldsymbol{X})$, [20,22]. Minimizing $H(C|\boldsymbol{X})$ should be desirable for learning classification models because we minimize the uncertainty remaining in the class variable once the value of $\boldsymbol{X}$ is known. Additionally, CLL is also related to the Kullback-Leibler divergence between the empirical and the model posterior class distribution [20,22].

On the other hand, it is quite usual to restrict the model complexity when learning Bayesian network classifiers. For instance, in this work, we use naive Bayes, and TAN models, which are Bayesian network classifiers with restricted network complexity. However, the model underlying the dataset (or the model which generated the data) may be more complex than the classifier that we use to model the dataset. Therefore, the estimation of both CLL (first term in Equation 7) and marginal log-likelihood (second term in Equation 7) is biased by the complexity restrictions that we set when learning the Bayesian network classifier. Hence, if the assumptions (or the restrictions) in the model that we learn are true, that is, the classification model learned from a dataset is close to the one that has generated this dataset, generative learning may present a good performance since it is able to model the relations between variables. Therefore, as generative learning is computationally more efficient, it may be preferred. By contrast, when the learned model is different from the *true* model, generative learning should perform worse than discriminative learning [16] because the bias for the generative model is higher.

In this section we present several experiments with both synthetic and real dataset from UCI [23] in order to illustrate the ideas introduced above. The Bayesian network classifiers used in the experiments neither deal with continuous variables nor missing values. Therefore, continuous variables have been discretized using Fayyad and Irani's method [24] and data samples containing missing values were ignored.

3.1 Synthetic Datasets

As was pointed before, it is thought that if the modeling assumptions (or restrictions) that we set when learning the Bayesian network classifier from data

are incorrect, a discriminative approach should be preferred because the bias is smaller. In this section we attempt to illustrate this behavior by learning classification models from datasets which have been sampled from random models with different structural complexity (NB, TAN and $p(C, \boldsymbol{X})$ models) and where the number of predictive variables vary in {8,12,16}, each predictive variable takes up to three states and the number of classes vary in {2,3}. For each configuration, we generate 10 random models and each one of these models is sampled 50 times to obtain 50 different datasets with 40,80,120,200,300,400 and 500 samples. This process yields 500 different datasets for each model configuration and dataset size. The datasets are used to learn NB, TAN and cTAN classifiers using both generative and discriminative approaches to learn the parameters (Section 2). The models used in these experiments may seem quite simple to the reader since the number of variables is not very high. The computational resources required to deal with the joint probability model prevent us for using more variables. Nevertheless, we think that these models are able to illustrate the performance of both generative and discriminative learning approaches.

Figure 1 shows how naive Bayes, TAN and cTAN models learned with both discriminative and generative parameter learning approaches perform in datasets sampled from random naive Bayes models. We can see in the plots that a naive Bayes model learned with a generative parameter learning approach performs better than the rest of the models. This is because a naive Bayes is enough to capture all the relations between variables. By contrast, TAN and cTAN models are overestimating the relations between variables, they create artificial relations that are not really represented in the dataset and that may lead them to obtain a worse classification rate than the naive Bayes model. Additionally, discriminative learning of the parameters for TAN models seems to perform slightly better than generative learning because the relations between variables assumed by the TAN model are not true in this case. However, we can not appreciate relevant differences between discriminative and generative structural learning.

Similarly, Figure 2 shows the results for datasets sampled from random TAN models. In this case, naive Bayes can not capture the relations between predictive variables. Hence, naive Bayes models obtain a worse classification rate than TAN and cTAN models. The performance of TAN and cTAN models is, again, very similar when the parameters are learned by a generative method, but a discriminative learning of the parameters for cTAN models (that is, discriminative learning of both structure and parameters) obtains, in this experiment, a worse classification rate. This is something surprising because the difference between generative and discriminative learning is significant for cTAN but not for TAN models. We think that the CLL function for cTAN models (when they are learned from a dataset sampled from TAN models) may present many local maxima where the TM algorithm can be trapped and thus the resultant model obtains a poor classification rate.

In general, we can see that, as expected, the generative learning performs better (or very similar) than the discriminative approach when the restrictions in the model that we are learning agree with the model used to generate the dataset.

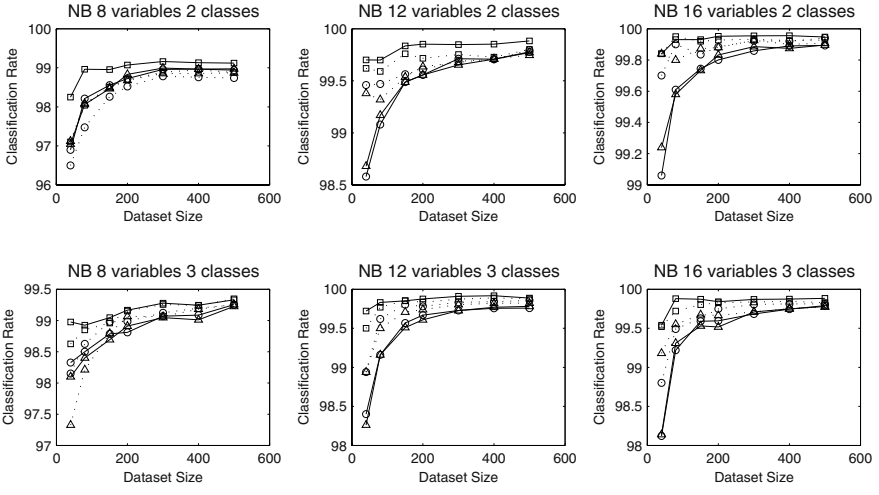


Fig. 1. Experiments with datasets sampled from random NB. Solid lines represent generative learning while dotted lines discriminative learning. □ denotes NB models; △ TAN models and ○ cTAN models. Each point in the plot represents the classification rate on average over 500 different datasets.

On the other hand, we would also like to test how naive Bayes, TAN and cTAN classifiers behave when the datasets are generated by using more complex models such as joint probability distributions, $p(C, \boldsymbol{X})$ (see Figure 3). In this experiment, the structural restrictions of the models that we are learning from the datasets (naive Bayes, TAN and cTAN) do not agree with the model that generate the data. Therefore, discriminative learning performs better, in terms of classification rate, than generative learning, at least for the parameter learning. However, the discriminative learning of TAN structures (cTAN models) performs very similar to the generative learning of the structure (TAN models).

3.2 UCI Repository

In this section, we develop a simple experiment that illustrates the use of the LL and CLL to guide the learning process of the parameters for a naive Bayes, TAN and cTAN models in real problems obtained from UCI repository.

Once the structure of the model is learned with the corresponding method described in Section 2, we obtain ten thousand different parameter sets at random and evaluate the LL, CLL and the classification rate of each classifier in the datasets. Then, we plot LL vs. classification rate and CLL vs. classification rate to evaluate the tendency of both LL and CLL scores with respect to the classification rate. Due to lack of space, we only show results for Breast, Hepatitis, Chess and Flare datasets but results for other UCI datasets are available at `www.sc.ehu.es/ccwbayes/members/guzman/discVSgen`.

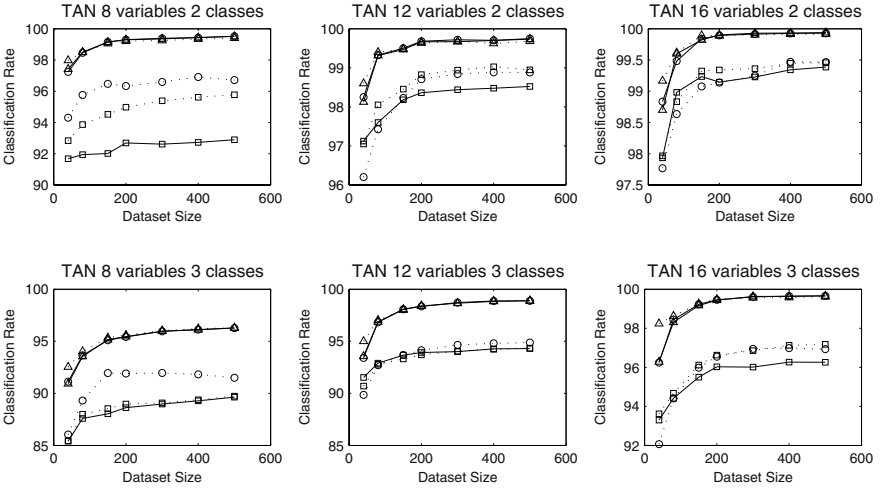


Fig. 2. Experiments with datasets sampled from random TAN. Solid lines represent generative learning while dotted lines discriminative learning. □ denotes NB models; △ TAN models and ∘ cTAN models. Each point in the plot represents the classification rate on average over the 500 different datasets.

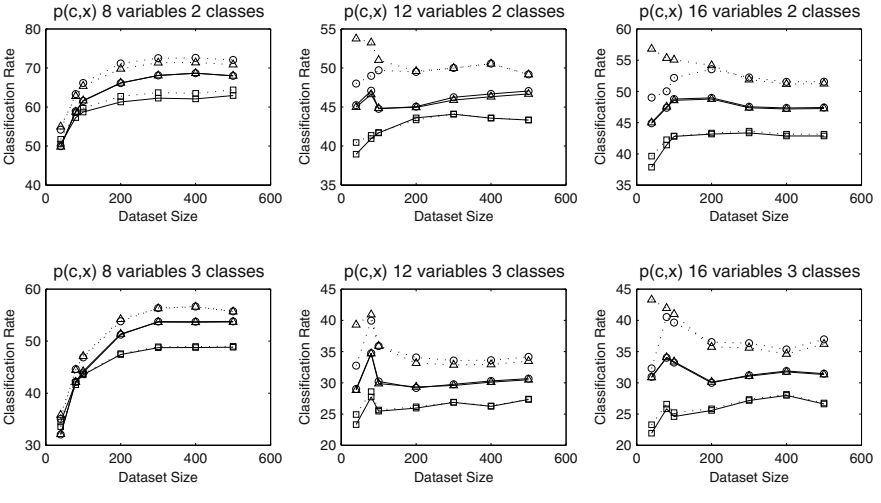


Fig. 3. Experiments with datasets sampled from random joint probability distributions. Solid lines represent generative learning while dotted lines discriminative learning. □ denotes NB models; △ TAN models and ∘ cTAN models. Each point in the plot represents the classification rate on average over the 500 different datasets.

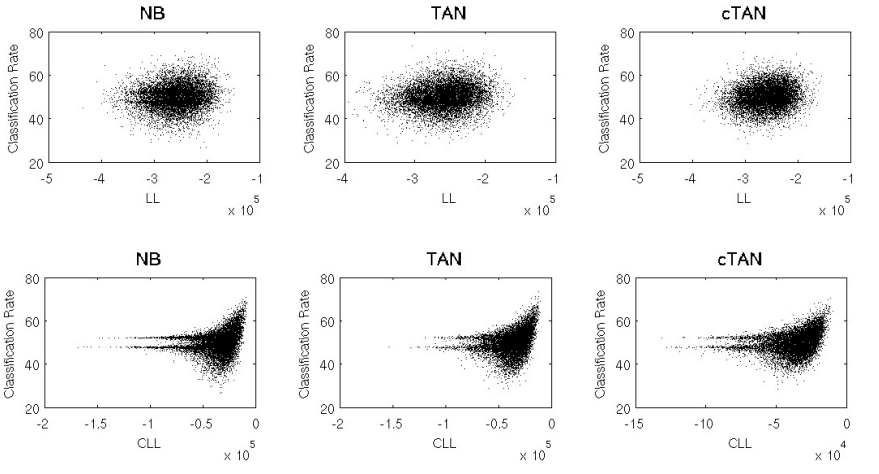


Fig. 4. Plot of the relation between LL, CLL and classification rate for Chess dataset

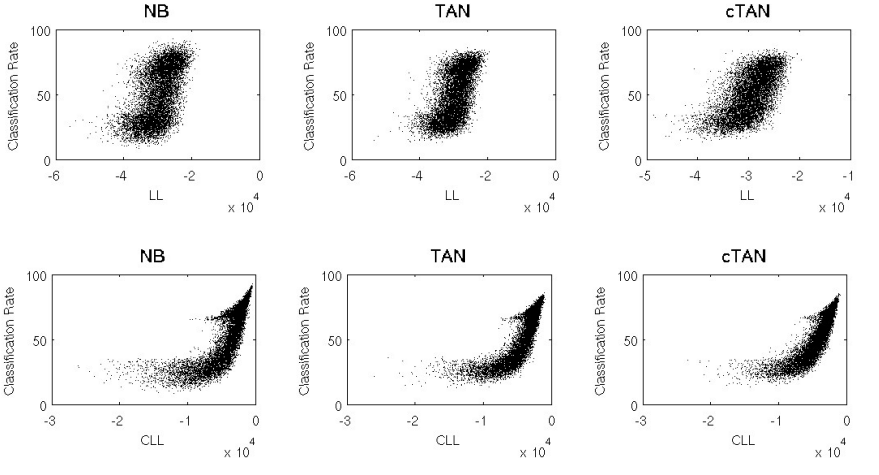


Fig. 5. Plot of the relation between LL, CLL and classification rate for Breast dataset

Figure 4 shows how the LL score is not related to the classification rate. However, in the same dataset, some correlation appears between CLL and classification rate. On the other hand, in Figures 5 and 6 it can be seen that there is some unclear relationship between LL and classification rate but CLL is much more correlated to the classification rate than LL. In fact, we can clearly appreciate in Figures 5 and 6 that as the CLL value increases, classification rate also increases. Nevertheless, other datasets from UCI such as Flare (Figure 7) present a clear relationship between LL and classification rate, but they also present a similar or stronger relationship between CLL and classification rate. From these

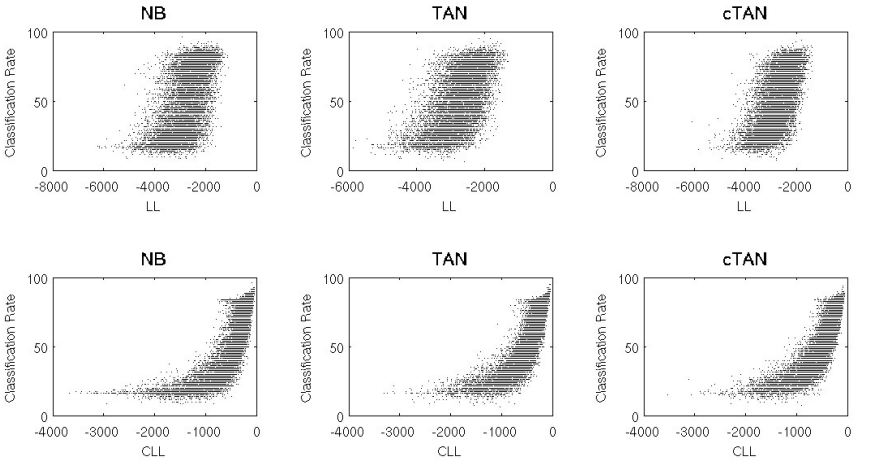


Fig. 6. Plot of the relation between LL, CLL and classification rate for Hepatitis dataset

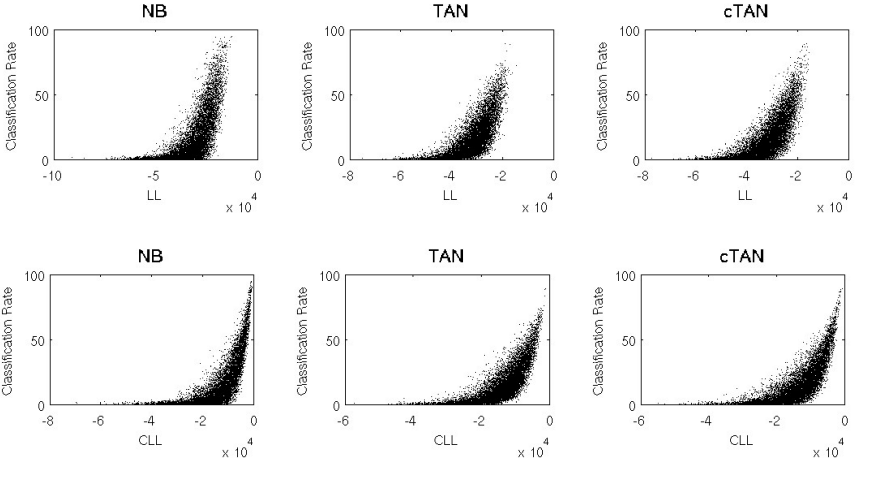


Fig. 7. Plot of the relation between LL, CLL and classification rate for Flare dataset

experiments (see also supplementary content on the web site) we can observe that CLL is more related to the classification rate than LL. Hence, in general, CLL and therefore discriminative learning seems a better approach to learn the parameters of Bayesian network classifiers when we want to maximize the classification rate. Nonetheless, the discriminative learning of the structure of TAN models (cTAN) does not seem to contribute to obtain a better classification rate than the generative leaning (TAN models).

4 Conclusions

In this paper we present a comparison between generative and discriminative methods to learn simple Bayesian network classifiers such as naive Bayes and TAN models. We empirically evaluate these methods in both synthetic and real datasets obtained from UCI repository in order to illustrate when discriminative learning is preferred. With these experiments, we corroborate some ideas stated by the machine learning community about generative and discriminative learning. Therefore, we show how, although it depends on the dataset of every specific problem, CLL score is preferred to LL score to learn, at least, the parameters of Bayesian network classifiers. Ideally, when the model underlying the dataset is simpler than the model we learn, the generative approach should be preferred since it is computationally more efficient and obtains a better or similar classification rate than the discriminative approach. However, when the model underlying the dataset is complex, discriminative learning seems a better choice. Obviously, this recommendation is difficult to follow because the model underlying the dataset for a real problem is usually unknown.

Acknowledgments

This work was supported by the SAIOTEK and Etortek research projects from the Basque Government, by the Spanish Ministerio de Educación y Ciencia under grant TIN 2005-03824 and by the Government of Navarra with a PhD grant awarded to the first author. Computing resources have been provided by the SGI-UPV (supported by Fondo Social Europeo, MCyT and Basque Government)

References

1. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco (1988)
2. Neapolitan, R.E.: Learning Bayesian Networks. Prentice-Hall, Englewood Cliffs (2003)
3. Larrañaga, P., Lozano, J.A., Peña, J.M., Inza, I.: Editorial. Machine Learning 59(3), 211–212 (2005)
4. Duda, R., Hart, P.: Pattern Classification and Scene Analysis. John Wiley and Sons, Chichester (1973)
5. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian networks classifiers. Machine Learning 29, 131–163 (1979)
6. Sahami, M.: Learning limited dependence Bayesian classifiers. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp. 335–338 (1996)
7. Cheng, J., Greiner, R.: Learning Bayesian belief network classifiers: Algorithms and system. In: Proceedings of the Fourteenth Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence, pp. 141–152 (2001)
8. Greiner, R., Zhou, W., Su, X., Shen, B.: Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. Machine Learning 59(3), 297–322 (2005)

9. Roos, T., Wettig, H., Grünwald, P., Myllymäki, P., Tirri, H.: On discriminative Bayesian network classifiers and logistic regression. Machine Learning 59(3), 267–296 (2005)
10. Feelders, A., Ivanovs, J.: Discriminative scoring of Bayesian network classifiers: A comparative study. In: Proceedings of the Third European Workshop on Probabilistic Graphical Models, pp. 75–82 (2006)
11. Santafé, G., Lozano, J.A. Larrañaga, P.: Discriminative learning of Bayesian network classifiers via the TM algorithm. In: Proceedings of the Eighth European Conference on Symbolic and Quantitative Approaches to Reasonig with Uncertainty, pp. 148–160 (2005)
12. Grossman, D., Domingos, P.: Learning Bayesian network classifiers by maximizing conditional likelihood. In: Proceedings of the Twentyfirst International Conference on Machine Learning, pp. 361–368 (2004)
13. Santafé, G., Lozano, J.A., Larrañaga, P.: Aprendizaje discriminativo de clasificadores Bayesianos. Revista Iberoamericana de Inteligencia Artificial (in Spanish) 29(10), 39–47 (2006)
14. Guo, Y., Greiner, R.: Discriminative model selection for belief net structures. In: Proceedings of the Twentieth National Conference on Artificial Intelligence (2005)
15. Pernkopf, F., Bilmes, J.: Discriminative versus generative parameter and structure learning of Bayesian network classifiers. In: Proceedings of the Twenty-Second International Conference on Machine Learning, pp. 657–664 (2005)
16. Rubinstein, Y.D., Hastie, T.: Discriminative vs. informative learning. In: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, pp. 49–53 (1997)
17. Vapnik, V.: Statistical Learning Theory. John Wiley and Sons, Chichester (1998)
18. Ng, A.Y., Jordan, M.I.: On discriminative vs. generative classifiers: A comparison of logistic regression and naïve Bayes. In: Proceedings of the Sixteenth Advances in Neural Information Processing Systems 14 (2002)
19. Bilmes, J.: Dynamic Bayesian multinets. In: Proceedings of the Sixteenth International Conference of Uncertainty in Artificial Intelligence, pp. 38–45 (2000)
20. Perez, A., Larrañaga, P., Inza, I.: Information theory and classification error in probabilistic classifiers. In: Todorovski, L., Lavrač, N., Jantke, K.P. (eds.) DS 2006. LNCS (LNAI), vol. 4265, pp. 347–351. Springer, Heidelberg (2006)
21. Edwards, D., Lauritzen, S.L.: The TM algorithm for maximising a conditional likelihood function. Biometrika 88(4), 961–972 (2001)
22. Cover, T.M., Thomas, J.A.: Elements of Information Theory, 2nd edn. John Wiley & Sons, Chichester (2006)
23. Blake, C., Merz, C.: UCI repository of machine learning databases (1998), `http://www.ics.uci.edu/~mlearn`
24. Fayyad, U., Irani, K.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, pp. 1022–1027 (1993)

PADUA Protocol: Strategies and Tactics

M. Wardeh, T. Bench-Capon, and F. Coenen

Department of Computer Science,
The University of Liverpool,
Liverpool L69 3BX, UK
{maya,tbc,frans}@csc.liv.ac.uk

Abstract. In this paper we describe an approach to classifying objects in a domain where classifications are uncertain using a novel combination of argumentation and data mining. Classification is the topic of a dialogue game between two agents, based on an argument scheme and critical questions designed for use by agents whose knowledge of the domain comes from data mining. Each agent has its own set of examples which it can mine to find arguments based on association rules for and against a classification of a new instance. These arguments are exchanged in order to classify the instance. We describe the dialogue game, and in particular discuss the strategic considerations which agents can use to select their moves. Different strategies give rise to games with different characteristics, some having the flavour of persuasion dialogues and other deliberation dialogues.

1 Introduction

In this paper we describe an approach to classifying objects in a domain not governed by strict rules which makes use of a novel combination of argumentation and data mining techniques. Our scenario is that classification is performed by two agents, each of which has their own set of records of past examples recording the values of a number of features presumed relevant to the classification and the correct classification. One of the agents will propose a classification, and a set of justifying reasons for the classification. This proposal is based on the application of an association rule mined from the agent's set of examples to the case under consideration. The classification is the consequent of the rule, and the antecedent gives the justifying reasons. The other agent will then use its set of examples to play "devil's advocate" and attempt to overturn the proposed classification. We call our system *PADUA (Protocol for Argumentation Dialogue Using Association Rules).*

This interaction can be viewed as a form of dialogue game (e.g. [5]), based on the exchange of arguments. Dialogue games come in a variety of flavours [11], including *Persuasion*, where each participant tries to persuade the other participant of its own thesis, by offering arguments that support this thesis, and *Deliberation*, in which the participants exchange arguments to reach an agreed decision, with neither of them committed to a particular position at the outset.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 465–476, 2007.

Our interaction has aspects of both, with the balance different according to the dialogue strategies employed. *Formal Dialogue Games* [7] are interactions between two or more players, where each player moves by making utterances, according to a defined set of rules known as a *Dialogue Game Protocol*, which gives the set of moves possibly expected after a previous move; choosing the best move among these moves is the *Strategy Problem*.

As mentioned above, the key idea of PADUA is to form arguments directly from some set of records providing examples relating to a particular domain, avoiding any need for expert analysis of the data, or knowledge representation. The repository of background knowledge used by each participant can be considered to be a binary valued data set where each record represents a previous case and each column an attribute taken from the global set of attributes described by the background knowledge. Given this set up we can apply *Association Rule Mining* (ARM) [1] techniques to the data set to discover relations between attributes, expressed in the form of *Association Rules* (ARs). In order to use this information, we follow the notion of presumptive argumentation as the instantiation of argument schemes subject to challenge through characteristic critical questions introduced by Walton [11]. PADUA makes use of a custom argument scheme and associated critical questions. In this paper we shall discuss the strategy problem in PADUA, and the consequences of the strategy used for the dialogue type.

The rest of this paper is organized as following: Section 2 describes the argument scheme and the basic structure of the PADUA protocol. Section 3 gives some necessary background on strategies in dialogue systems. Section 4 discusses in detail the suggested strategy heuristics to be applied in PADUA protocol. Section 5 gives a detailed example of the suggested strategy, and some discussion of the relation between these strategies and dialogue types.

2 PADUA Protocol

The model of argumentation we will follow is that of [11] in which a prima facie justification is given through the instantiation of an argument scheme. This justification is then subject to a critique through a number of critical questions which may cause the presumptive conclusion to be withdrawn.

The basic argument scheme is one we have devised for the purpose, Argument from Proposed Rule. The *Premises* are:

1. *Data Premise*: There is a set of examples D pertaining to the domain.
2. *Rule Premise*: From D a Rule R can be mined with a level of confidence greater than some threshold T. R has antecedents A and a conclusion which includes membership of class C.
3. *Example Premise*: Example E satisfies A.
4. *Conclusion*: E is a C because A.

This can be subject to a number of critical questions:

- *Can the case be distinguished from the proposed rule*? If D supports another Rule R2 and the antecedents of R2 subsume those of R, and are satisfied by E, and the confidence of R2 is below T, this suggests that these additional features may indicate we are dealing with some kind of exception to R.
- *Does the rule have unwanted consequences*? If the conclusion of R includes some fact F not satisfied by E, this suggests that R is not applicable to E.
- *Is there a better rule with the opposite conclusion*? If there is another rule R3 which can be mined from D and the antecedents of R3 are satisfied by E, and the confidence of R3 is greater than that of R, this suggests that R is not applicable to E.
- *Can the rule be strengthened by adding additional antecedents*? This does not challenge the classification, but rather the justification for the classification. If there is another Rule R4 which can be mined from D and the antecedents of R4 subsume those of R, and are satisfied by E, and the confidence of R4 is greater than that of R, this suggests that the additional features should be included in the justification of the classification.
- *Can the rule be improved by withdrawing consequences*? This challenges neither the classification, nor the justification, but the rule proposed. If there is another Rule R5 which can be mined from D and the conclusions of R subsume those of R5 and include a feature not satisfied by E, provided the confidence of R5 remains above the threshold, this suggests that the additional features should be excluded from the rule justifying the classification.

This argument scheme and these critical questions form the basis of the PADUA dialogue game.

2.1 Dialogue Scenario

The proposed dialogue game consists of two players (the proponent and the opponent) which have conflicting points of views regarding some case (C). The proponent claims that the case falls under some class (c_1), while the opponent opposes the proponent's claim, and tries to prove that case actually falls under some other class ($c_2 = \neg c_1$). Each player tries to establish its point of view by the means of arguments based on association rules, which are mined from player's own database, using an association rule mining technique as described in [4].

The proponent starts the dialogue by proposing some AR ($R_1 : P \rightarrow Q$), to instantiate the argument scheme. The premises (P) match the case, and the conclusion (Q) justifies the agent's position. Then the opponent has to play a legal move that would undermine the initial rule proposed by the proponent: these moves are based of the five critical questions described above. As can be seen from the questions, four of these moves involve some new rule. This is mined from the opponent's background database, and represents an attack on the original rule. The turn then goes back to the proponent which has to reply appropriately to the last move. The game continues until one player has no adequate reply. Then this player loses the game, and the other player wins.

2.2 PADUA Framework

The formal framework we suggest *Argumentation Dialogue Framework* (ADF) is defined as follows:

$$ADF =< P, Attr, C, M, R, Conf, playedMoves, play > \quad (1)$$

Where P: denotes the players of the dialogue game. $Attr$: denotes the whole set of attributes in the entire framework. C: denotes the case argued about. M: denotes the set of possible (legal) moves. R: denotes the set of rules that govern the game. $Conf$: denotes the confidence threshold, all the association rules proposed within this framework must satisfy this threshold. $playedMoves$: denotes the set of moves played in the dialogue so far, this set of played moves represents the commitment store of the dialogue system under discussion. Finally, $play$: is a function that maps players to some legal move.

2.3 PADUA Players

Each player in PADUA game ($\forall p \in P = Pro, Opp.$)is defined as a dialogical agent [3]:

$$\forall p \in P : p =< name_p, Attr_p, G_p, \Sigma_p, >>_p > \quad (2)$$

where: $name_p$: is the *player* (*agent*) name, here: $\forall p \in P$ then $name(p) \in \{pro, opp\}$. $Attr_p$: is the set of attributes this player can understand. G_p: is the set of *goals* this player tries to achieve, here G_p is defined as a subset of the attributes set $Attr_p$, i.e. G_p is the set of attributes (classes) this player tries to prove true. Σ_p: is the set of ARs the player has mined from its background database, hence fore p is defined as follows: $\forall p \in P : \Sigma_p = \{r_1 \dots r_m\}$, where $r_i =< Prem, Con, Conf >$ is an association rule and can be read as $Prem \rightarrow Conc$ with a confidence $= Conf$. The elements of $Prem$ and $Conc$ are defined as a tuple $< attribute, value >$, where $attribute \in Attr_p$, and $value$ is the list of values assigned to this attribute in the given rule. $>>_p$: represents the preferences order over Σ_p, a definition of this preference relation ship is suggested as $>>_p$: $\Sigma_p \times \Sigma_p \rightarrow \{true, false\}$, but the exact implementation of this relation may differ from player to player.

2.4 PADUA Legal Moves

The set of moves (M) consists of 6 possible moves, one based on instantiating the argument scheme and five based on the critical questions against an instantiation. They are identified as follows:

1. *Propose Rule*: p plays this move to propose a new rule with a confidence higher than some confidence threshold.
2. *Distinguish*:this move is played to undermine a previously played move, as it adds some new premise(s) to this rule, such that the confidence of the new rule is lower than the confidence of the original rule (and/or lower than some acceptance threshold).

3. *Unwanted Consequences*: Here p suggests that certain consequences (conclusions) of some rule do not match the case under discussion.
4. *Counter Rule*: p plays this move to propose a new rule that contradicts the previous rule. The confidence of the proposed counter rule should be higher than the confidence of the previous rule (and/or than the threshold $Conf$).
5. *Increase Confidence*: p plays this move to add some new premises to a previous rule so that the overall confidence rises to some acceptable level.
6. *Withdraw Unwanted Consequences*: p plays this move to exclude the unwanted consequences of the rule it previously proposed, while maintaining a certain level of confidence.

This defines the formal dialogue game. We now consider the strategies that might be used when playing the game. First we consider some related previous work on developing strategies for formal dialogue games.

3 Dialogue Strategies: Background

This section discusses some previous argumentation systems that have considered argument selection strategies:

Moore, in his work with the DC dialectical system [8], concluded from his studies that an agent's argumentation strategy is best analyzed at three levels:

1. Maintaining the focus of the dispute.
2. Building its point of view or attacking the opponent's one.
3. Selecting an argument that fulfils the objectives set at the previous two levels.

The first two levels refer to the agent's strategy, i.e. the high level aims of the argumentation, while the third level refers to the tactics, i.e. the means to achieve the aims fixed at the strategic levels. Moore's requirements form the basis of most other research into agent argumentation strategies.

In [2] a computational system was suggested that captures some of the heuristics for argumentation suggested by Moore. This system requires a preference ordering over all the possible arguments, and a level of prudence to be assigned to each agent. The strength of an argument is defined according to the complexity of the chain of arguments required to defend this argument from the other arguments that attack it. An agent can have either a "build" or a "destroy" strategy. When using the build strategy (b-strategy), an agent tries to assert arguments the strength of which satisfies its prudence level. If the b-strategy fails, it switches to the destroy strategy (d-strategy), where it tries to use any possible way to attack the opponent's arguments. The basic drawback of this approach is that computational limits may affect the agent's choice.

In [6] a three layer system was proposed to model argumentation strategies: the first layer consists of the "default" rules, which have the form (utterance - condition); the higher two layers provide preference orderings over the rules. The system is shown to be deterministic, i.e. a particular utterance is selected

in a given situation every time, but this system still requires hand crafting of the rules.

In [10], a decision heuristic was proposed to allow the agents to decide which argument to advance. The idea behind this work is that an agent should, while attempting to win a dispute, reveal as little of what it knows as possible, as revealing too much information in a current dialogue might damage an agent's chances of winning a future argument. A new argumentation framework was developed to represent the suggested heuristics and arguments. The main shortcoming of this approach is the exponential complexity of the algorithms used.

4 Strategies and Tactics for PADUA

In PADUA, a player $p \in P$ must select the kind of move to be played, and also the particular content of this move depending on: the thesis this player aims to prove true (or false), the case under discussion, the player's set of association rules, the amount of information this agent is willing to expose in its move, and the player's current state in the dialogue. All these factors must be considered in the strategy the player adopts and the tactics applied to implement this strategy.

Table1 lists the *possible next moves* after each of the legal moves in PADUA protocol. A player must select a single move to play in its turn; moreover every possible next move is associated with a set of possible rules: this set contains the rules that match the selection criteria of the move, i.e. their confidence, premises and conclusion match this move. Except for *unwanted consequences*, the moves introduce a new rule. Proposing a counter rule leads to a switch in the rule being considered, so entering a nested dialogue. The notion of move (act) and

Table 1. Possible Moves

Move	*Next Move*	*New Rule*
1	2,3,4	yes
2	1,3,5	yes
3	1,6	No
4	1,2,3	Nested Dialogue
5	2,3,4	yes
6	2,3,4	yes

content selection is argued to be best captured at different levels, as suggested by Moore [8]. In [2] the first level of Moor's layered strategy was replaced with different profiles for the agents involved in the interaction. We also adopt this approach. Here we also add another level to Moore's structure (level 0) which distinguishes PADUA games into two basic classes. In one players attempt to win using as few steps as possible, i.e. the move's type and content are chosen so that the played move gives the opponent's the least freedom to plan its next move. In the other, games that are played to fully explore the characteristics of the underlying argumentation system, and dialogue game, so here the move's

type and content are chosen so that the played move will restrict the opponent's freedom to plan its next move to the least extent possible. The layered strategy system we adopt is defined as follows:

- Level 0: *Define the game mood*: i.e. *Win mode or Dialogue mode*.
- Level 1: *Define the players (agents) profiles*.
- Level 2: *Choose to build or destroy*: where in a *Build mode* the player tries to build its own thesis, while in a *Destroy mode* the player tries to destroy the opponent's thesis.
- Level 3: *Choose some appropriate argumentative content*: depending on the tactics and heuristics suggested.

4.1 Agent Profile

In [3], which used arguments based on standard if then rules, five classes of agent profiles were defined as follows:

1. *Agreeable Agent*: Accept whenever possible.
2. *Disagreeable Agent*: Only accept when no reason not to.
3. *Open-minded Agent*: Only challenge when necessary.
4. *Argumentative Agent*: Challenge whenever possible.
5. *Elephant Child Agent*: Question when ever possible.

In this paper we consider only the first two profiles (i.e. agreeable and disagreeable agents), as these attitudes are the most appropriate for the particular argument scheme we are using.

4.2 PADUA Strategy

The function *Play* is defined as follows:

$$Play : P \times M_{poss} \times R_{poss} \times playedMoves \times S \times \rightarrow M \tag{3}$$

Where: P is the set of game players; $playedMoves$ is the set of moves played in the dialogue so far; and M is the set of possible (legal) moves. M_{poss}: is the set of the possible moves this player can play $M_{poss} \subseteq M$ (as defined in Table1). R_{poss}: is the set of legal rules that this agent can put forward in the dialogue ($R_{poss} \subseteq 2^{\Sigma_p}$); this set contains the rules that match the each of the possible moves. S: is the *Strategy Matrix*, and has the form $S = [g_m, profile_P, s_m]$ where: $g_m \in G_m$: is the game mode, where $G_m = \{win, dialogue\}$, $profile_P \in Profile_P$: is the player profile, where $Profile_P = \{agreeable, disagreeable\}$, and finally, $s_m \in S_m$: is the strategy mode, where $S_m = \{build, destroy\}$.

4.3 PADUA Tactics

A set of tactics are suggested to fulfil the strategic considerations discussed above; these concern the best move to play and, where applicable, the content of the chosen move, i.e. the best rule to be put forward in the dialogue.

Legal Moves Ordering. *Legal moves' ordering* defines the order in which legal (possible) moves are considered when selecting the next move. All games begin with Propose Rule: there are three possible responses to this, and these in turn have possible responses. The preference for these moves depends on whether the agent is following a build or a destroy strategy. In a destroy strategy the agent will wish to discredit the rule proposed by its opponent, and hence will prefer moves such as *unwanted consequences* and *distinguish*. In contrast when using a build strategy an agent will prefer to propose its own rule, and will only attempt to discredit its opponents rule if it has no better rule of its own to put forward. The preferred order for the two strategies is shown in Table2.

Whether players are agreeable or disagreeable will have an influence on whether the agent wishes to dispute the rule put forward by its opponent, and, the nature of the challenge if one is made.

Table 2. Possible Moves Preferences

Last Move	*Build Mode*	*Destroy Mode*
1	4,3,2	3,2,4
2	1,3,5	3,5,1
3	1,6	6,1
4	1,3,2	3,2,1
5	1,3,2	3,2,1
6	1,3,2	3,2,1

Agreeable Players. An agreeable player $ap \in P$ accepts a played rule without challenging it if:

1. An exact match of this rule can be found in its own set of association rule (Σ_{ap}) with a higher or similar confidence.
2. Can find partial match of this rule in its own set of association rule (Σ_{ap}), a rule $r_{pm} \in \Sigma_{ap}$ is considered to be a partial match of another rule $r \in \Sigma_{ap}$ if it has the same conclusion (consequences) of r, it's set of premises is a superset of rule r premises, and all these premises match the case; and finally it has a higher or similar confidence.

Otherwise the agreeable agent challenges the played move, depending whether it wishes to build or destroy using the legal moves preferences shown in Table2 selecting a rule using the following content tactics:

1. *Confidence*: Confidence of moves played by agreeable agent should be **considerably** lower/higher than the attacked rule, otherwise the agent agrees with its opponent.
2. *Consequences*: Consequences always contain a class attribute.
   Minimum changes to previous move consequences.
   As few attributes as possible.
3. *Premises*: Premises are always true of the case.
   Minimum changes to previous move premises.
   As few attributes as possible.

Disagreeable Players. A disagreeable agent accepts a played rule if and only if all possible attacks fail, and so does not even consider whether its data supports the rule; the choice of the attack (i.e legal move) to be played depends on the preferences shown in Table2 and the choice of rule is in accordance with the following content tactics:

1. *Confidence*: Confidence of moves played can be:
   (a) Considerably different from last move
   (b) Slightly different from last move.
   The choice of confidence depend on the general mode of game whether it's in a win-mood or a dialogue-mood.
2. *Consequences*: Consequences always contain a class attribute.
   As few attributes as possible.
3. *Premises*: Premises are always true of the case.
   As few attributes as possible.

Best Move. Table 3 brings these considerations together and shows the best move relative to the agent type and the game mode, for each of the move types. For example in win mode an agent will want to propose a rule with high confidence, as one which the opponent is likely to be forced to accept, whereas in game mode, where a more thorough exploration of the search space is sought, any acceptable rule can be used to stimulate discussion.

Table 3. Best move content tactics

Best Moves				
	Agreeable		**Disagreeable**	
	Win mode	Game mode	Win mode	Game mode
propose	High confidence	Average confidence	High confidence	Average confidence
	Fewest attributes	Average attributes	Fewest attributes	Fewest attributes
distinguish	Lowest confidence	Average drop	Lowest confidence	Average drop
	Fewest attributes	Fewest attributes	Fewest attributes	Fewest attributes
Unwanted consequences	If some consequences are not in or contradict the case	Only if some consequences contradict the case	If some consequences are not in or contradict the case	
Counter rule	Average confidence	High confidence	High confidence	Average confidence
	Fewest attributes	Fewest attributes	Average attributes	Fewest attributes
Increase Confidence	Highest confidence	Average increase	Highest confidence	Average increase
	Fewest attributes	Fewest attributes	Fewest attributes	Fewest attributes
Withdraw unwanted consequences	The preferable reply to unwanted consequences attack → selecting criteria is the same of the very last move that led to the unwanted consequences.			

5 Example

Our example domain concerns the voting records of US members of Congress, on the basis of which we wish to classify them according to party affiliation. Although there will be typical Democrat and Republican views on various issues, people may vote against the party line for personal or regional reasons. Some members of Congress may be more maverick than others. Thus, while there is no defining issue which will allow us to classify with certainty, we can argue for a classification on the basis of voting records. The data set we use is taken from [9], and it represents the U.S. House of Representatives members of Congress (in the 1984 US congressional elections) on the 16 key votes identified by the (CQA). The congressional voting records database contains 435 instances, among which (45.2%) are Democrats and (54.8%) are Republicans. The dataset original 17 binary attributes (including the class attribute) were normalized to 34 unique numerical attributes, each corresponds to certain attribute value. This dataset was horizontally divided into two equal size datasets, each of which was assigned to a player in PADUA framework. Rules were mined from this dataset using 30% support, and 70% confidence thresholds.

We have experimented by running several PADUA dialogue games, starting from the same case. The difference between the games lays in the underlying strategy options of each agent that participate in each of these games.

Table 4 shows the attributes of the case used in the example.

Table 4. Example Case

Case:		[5, 7, 13, 15, 17, 21, 24, 26, 29]	
5:	adoption-of-the-budget - resolution=y.	7:	physician-fee-freeze=n.
13:	anti-satellite- test-ban=y.	15:	aid-to-nicaraguan -contras=y.
17:	mx-missile=y.	21:	synfuels-corporation-cutback=y.
24:	education-spending=n.	26:	superfund-right-to-sue=n.
29:	duty-free- exports=y.		

As an illustration, we will describe the run with two disagreeable agents playing in win mode, the proponent (*Prop*) using a *build strategy* and the opponent (*Opp*) a *destroy strategy*. *Prop* begins by proposing a rule to justify saying that the member of Congress concerned is a Democrat: **R1**: Democrat because `education-spending=n` and **duty-free-exports =y** with a (97.66%) confidence. *Opp* can reply by distinguishing this rule, since adding the premise **aid-to-nicaraguan-contras=y** reduces confidence to 80.43%. *Prop* now proposes a new rule: **R2**: Democrat because `mx-missile=y` and `duty-free-exports=y` with a (98.63%) conifednce. This rule cannot be distinguished or countered since there is no better rule for Republican and so *Prop* wins.

Note how, *Opp*, being in *destroy mode*, uses first the distinguish move and only proposes a rule if this move cannot be played. In *build mode Opp* plays a rule of its own. Note also that the distinction made greatly reduces the confidence, whereas a distinction with a less drastic effect could have been played in game mode. When *Opp* is an *agreeable agent* it would simply accepts the proposed rule, as it too can mine the rule with sufficient confidence. Where *Prop* is in destroy mode, it responds to the distinction with an increase confidence move, forcing *Opp* to propose a rule of its own.

As would be expected, in *game mode*, longer dialogues are produced. Where the agents are both agreeable, game mode leads to a series of rule proposals until a mutually acceptable one is found. Where *Opp* is in *destroy mode*, *Prop*'s proposals will be met by a distinction, and where *Opp* is in build mode it will produce counter proposals as long as it can. Where *Prop* is in destroy mode it will make use of the unwanted consequences move to refute *Opp*'s distinction if possible. Where both agents are disagreeable and in win mode, because the game does not terminate on the proposal of an acceptable rule, this last move, refuting a distinction by pointing to unwanted consequences which cannot be met with a withdraw consequences move, is what ends the game.

6 Discussion

Padua provides a way of determining the classification of cases on the basis of distributed collections of examples related to the domain without the need to share information, and without the need for analysis and representation of the examples. The argumentation leads to a classification which, while uncertain, is mutually acceptable and consistent with the different collections of examples.

Different strategies for move selection give rise to dialogues with different characteristics. Using disagreeable agents gives rise to a persuasion dialogue, since the opponent will do anything possible to avoid accepting the proposal. Win mode will lead to the swiftest resolution: game mode between disagreeable agents will lead to a lengthier exchange, and concession may be forced without the best argument being produced. A dialogue between two agreeable agents has the characteristics of a deliberation dialogue in that here the opponent is happy to concede once an acceptable proposal has been made. Win mode may be a very short exchange, since this simply verifies that Prop's best rule is also acceptable with respect to the second agent's data set. When game mode is used, the game has the flavour of brainstorming in that more ideas, even some which are less promising, will be explored.

Further work will empirically explore our system to examine the efficiency and quality of classifications and the effect of giving the individual data sets used by the agents particular characteristics. We also intend to explore domains in which classification is into a enumerated set of options rather than binary, and develop an extended version of the game with more than two participants.

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Association rules between sets of items in large databases. In: Proc. of ACM SIGMOD Int. Conf. on Management of Data, Washington, pp. 207–216 (May 1993)
2. Amgoud, L., Maudet, N.: Strategical considerations for argumentative agents (preliminary report). In: Proc. of 9th Int. Workshop on Non-Monotonic Reasoning (NMR), Toulouse, France, April 2002. Special session on Argument, Dialogue, Decision, pp. 409–417 (2002)
3. Amgoud, L., Parsons, S.: Agent dialogues with conflicting preferences. In: Proc. of 8th Int. Workshop on Agent Theories, Architectures and Languages, Seattle, Washignton, August 2001, pp. 1–15 (2001)
4. Coenen, F.P., Leng, P., Goulbourne, G.: Tree Structures for Mining Association Rules. Journal of Data Mining and Knowledge Discovery 8(1), 25–51 (2004)
5. Hamblin, C.L.: Fallacies. Methuen (1970)
6. Kakas, A.C., Maudet, N., Moraitis, P.: In: Rahwan, I., Moraïtis, P., Reed, C. (eds.) ArgMAS 2004. LNCS (LNAI), vol. 3366, Springer, Heidelberg (2005)
7. Mcburney, P., Parsons, S.: Games That Agents Play: A Formal Framework for Dialogues between Autonomous Agents. In Jo. of logic, language and information 11(3), 315–334 (2002)
8. Moore, D.: Dialogue game theory for intelligent tutoring systems. PhD thesis, Leeds Metropolitan University (1993)
9. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
10. Oren, N., Norman, T.J., Preece, A.: Loose Lips Sink Ships: a Heuristic for Argumentation. In: Proc. of 3rd Int. Workshop on Argumentation in Multi-Agent Systems (Argmas 2006), Hakodate, Japan, May 2006, pp. 121–134 (2006), Proceedings available at `http://homepages.inf.ed.ac.uk/irahwan/argmas/argmas06/argmas2006proceedings.pdf`
11. Walton, D.N., Krabbe, E.C.W.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. SUNY Press, Albany (1995)
12. Walton, D.N.: Argument Schemes for Presumptive Reasoning. Lawrence Erlbaum Associates, Mahwah (1996)

A Semi-naive Bayes Classifier with Grouping of Cases*

Joaquín Abellán, Andrés Cano, Andrés R. Masegosa, and Serafín Moral

Department of Computer Science and Artificial Intelligence.
University of Granada, Spain
{jabellan,acu,andrew,smc}@decsai.ugr.es

Abstract. In this work, we present a semi-naive Bayes classifier that searches for dependent attributes using different filter approaches. In order to avoid that the number of cases of the compound attributes be too high, a grouping procedure is applied each time after two variables are merged. This method tries to group two or more cases of the new variable into an unique value. In an emperical study, we show as this approach outperforms the naive Bayes classifier in a very robust way and reaches the performance of the Pazzani's semi-naive Bayes [1] without the high cost of a wrapper search.

1 Introduction

The naive Bayes classifier [14] is a probabilistic classifier that makes very strong independence assumptions and performs very well on many data sets. It simplifies the learning task by assuming that the attributes are independent given the variable to classify (no structural learning is required).

We shall use $\mathbf{X} = \{X_1, \ldots, X_n\}$ to denote features describing the instances to be classified and C for the class variable. The supervised classification problem under naive Bayes classifier assumptions reduces to find c^* such as:

$$c^* = arg_c \max P(C = c) \prod_i P(X_i = x_i | C = c)$$

There are many attempts to improve the accuracy of the naive Bayes classifier taking advantage of interdependence between attributes. These methods are collectively called *semi-naive Bayesian methods* [5]. A comparation of such methods can be found in [6]. Pazzani [1] proposes one of such methods. That classifier carries out a search into the whole set of attribute pairs with the aim of merging them and creating a new compound attribute with the Cartesian product of the value sets of the original attributes. A joining operation is carried out when the predictive accuracy, estimated by a leave-one-out process, of the naive Bayes classifier with the joined variables is better than the classifier without the joining operation. Pazzani also evaluates two greedy algorithms for the model

* This work has been supported by the Spanish 'Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía' and 'Ministerio de Educación y Ciencia', under Projects TIC-276 and TIN2004-06204-C03-02 and FPU scholarship (AP2004-4678).

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 477–488, 2007.

selection. A forward sequential selection and joining (FSSJ) scheme where each non-selected variable is evaluated in two different ways in each step: considering it joined with each one of the already selected or joined variables (in this case there are so many evaluations as selected variables in this step) and considering it independent of the selected ones given the class variable. In each step, the most promising operation, joining or addition, over a non selected variable in terms of accuracy is carried out. The backward sequential elimination and joining (BSSJ) scheme employs the same ideas but starting with all the variables and selecting the most promising operation between joining or removing over the non-removed variable set. In both cases, the procedure stops where there is no improvement with the possible operations. The experimental study shows a lighter outperformance of the BSSJ scheme but with a huge computation cost that makes this last methodology prohibited in many classification problems. The FSSJ scheme shows the best trade-off in terms of accuracy and efficiency.

In a posterior work, Domingos et al. [7] try to show that an entropy based metric for measuring the degree of dependence of the attributes is not a good criterion for joining variables. The used measure is taken from Kononenko [5]:

$$D(X_i, X_j|C) = H(X_i|C) + H(X_j|C) - H(X_i, X_j|C)$$

where H stands for the entropy with probabilities estimated from the relative frequencies in the learning sample D.

$D(X_i, X_j|C)$ measure is zero when X_i and X_j are completely independent given C and increases with their degree of dependence.

In an empirical study, they show as the semi-naive Bayes method of Pazzani [7] using this entropy based measure for the joining criterion, instead of the estimated accuracy, does not outperform the naive Bayes classifier in none of eleven UCI data sets in a significant way. They finally suggest that accuracy based metrics are better score for joining variables than metrics measuring the degree of dependence between attributes.

Nevertheless, the main problem of the Pazzani approach [1] is the high-cost associated to accuracy based metrics since a cross validation process is carried out at each step.

However, in a recent work, Abellán [2] successful employs uncertainty measures based on imprecise probabilities, such us maximum entropy [3,4], to decide the joining of two variables, showing that the used of new entropy based measures it is a plausible possibility for this kind of classifiers.

In this work, we propose and study three new filter measures to choose the variables to join, which are much less costly than Pazzani's criterion. One of the problems associated with the process of joining variables is that the number of possible values grows exponentially as a function of the number of merged variables. To avoid this, we also introduce a complementary grouping process for reducing the number of states of the new joined variables.

In an empirical study we show that our approach, based on the combination of joining variables and grouping cases, outperforms the naive Bayes classifier and reaches the performance of the Pazzani semi-naive Bayes procedure with a much less computational cost.

The rest of the paper is organized as follows. Section 2 describes the proposed classifier. Concretely, Section 2.1 describes the joining process and Section 2.2 the grouping method. In Section 3 the experimental results are showed. Finally, in Section 4 the conclusions and future developments are exposed.

2 The Semi-naive Bayes with Grouping Cases

Firstly, the process to join variables is exposed. Here, we have evaluated several metrics to decide about the joining of two variables. Then, a grouping process is described in order to reduce the number of states of the joined variables. Again, several metrics are proposed and evaluated. And finally, the whole steps of the classifier are depicted.

2.1 Joining Criterion

Following the same scheme of [1] and [5], all possible pairs of variables are considered at each step with a given metric. The metric evaluates the convenience of joining the two variables with respect to keeping them separated. In this way, the most suitable ones are merged by creating a new compound variable with the Cartesian product of the value sets of the original variables. This procedure is used in an iterative way: the old joined variables are removed and the new one is included as a candidate to be joined again with another variable. The process continues until there is not more suitable variable pairs to be joined.

In this work, we propose three filter metrics as a **joining criterion**. Each one has a **joining condition** (**JC**(X_i, X_j)) that tests whether the variables X_i and X_j can be joined and a **joining metric** (**JM**) that allows to select the most suitable pair that should be joined.

Bayesian Dirichlet equivalent Metric (BDe). Bayesian scoring criteria have been widely used to choose between several alternative models [8], because of the inherent penalty that they give to the more complex models in order to prevent against over-fitting.

The Bayesian scores measure the quality of a model, M_i, as the posterior probability of the model giving the learning data D. Usually the logarithm of this quantity is considered for computational reasons giving rise to:

$$Score(M_i : D) = \ln P(M_i|D) = \ln P(D|M_i) + \ln P(M_i) - \ln P(D)$$

This value can be computed under suitable hypothesis. The BDe (Bayesian Dirichlet equivalent) [8] supposes an uniform prior probability over the possible models and a prior Dirichlet distribution over the parameters with independence for the parameters of different conditional distributions. Usually a global sample size, S, is considered and then it is assumed that for each variable Z the prior probability about the vector $(P(z))_z$ is Dirichlet with the same parameters S/k_Z for all values $P(z)$, where k_Z is the number of possible values of Z.

The metric for joining attributes X_i and X_j is computed as the difference: $Score(M_1 : D) - Score(M_2 : D)$, where M_1 is a naive Bayes model in which X_i and X_j are joined and M_2 a model in which they are considered conditional independent given the class. Under global sample size S, this difference can be computed as:

$$\mathbf{JM}_{BDe}(X_i, X_j) = \sum_c \ln\left(\frac{\Gamma(S/k_C)}{\Gamma(S/k_C + N_c)}\right)\left(T_{C,X_i,X_j} - T_{C,X_i} - T_{C,X_j}\right)$$

where

$$T_{C,X_i,X_j} = \sum_{x_i,x_j}\left(\frac{\Gamma(S/(k_C.k_{X_i}.k_{X_j}) + N_{cx_ix_j})}{\Gamma(S/(k_C.k_{X_i}.k_{X_j}))}\right), T_{C,X_k} = \sum_{x_k}\left(\frac{\Gamma(S/(k_C.k_{X_k}) + N_{cx_k})}{\Gamma(S/(k_C.k_{X_k}))}\right)$$

$\Gamma(.)$ is the gamma function ($\Gamma(\alpha+1) = \alpha.\Gamma(\alpha)$), $N_{cx_ix_j}$ is the number of occurrences of $[C = c, X_i = x_i, X_j = x_j]$ in the learning sample D (analogously for N_c and N_{x_kc}).

We select the pair X_i, X_j with greatest metric and the attributes are merged if the joining condition is verified i.e.

$$\mathbf{JC}_{BDe} = [JM_{BDe} > 0]$$

The Expected Log-Likelihood Under Leaving-One-Out (L10). The score of a model M_i for a set of data D is obtained by adding for each vector of cases $(\mathbf{x}, c) \in D$, the logarithm of $P(c|\mathbf{x})$, where the probability P is obtained by estimating the parameters of M_i with $D - \{(\mathbf{x}, c)\}$. That is, an estimation of the log-likelihood of the class [9] is carried out with a wrapper leaving-one-out procedure.

The metric for joining attributes X_i and X_j is computed as the difference of scores between the model in which X_i and X_j are joined and the model in which they are considered conditional independent given the class. However, this value can depend on the remaining attributes and can be difficult to compute in a closed form. For that reason we compute it considering that only variables X_i and X_j and C are in the model. This can be considered as an approximation which allows a fast computation. This metric is computed as:

$$\mathbf{JM}_{L1O}(X_i, X_j) = \sum_{c,x_i,x_j} N_{cx_ix_j}\left[\ln\left(\frac{P^*(x_i, x_j|c)P^*(c)}{\sum_{c'} P^*(x_i, x_j|c')P^*(c')}\right)\right] -$$

$$-N_{cx_ix_j}\left[\ln\left(\frac{P^*(x_i|c)P^*(x_j|c)P^*(c)}{\sum_{c'} P^*(x_i|c')P^*(x_j|c')P^*(c')}\right)\right]$$

where the probabilities P^* are estimated from the sample using the Laplace correction and discounting 1 the absolute frequencies of values (c, x_i, x_j) in the sample:

$$P^*(x_i, x_j|c) = \frac{N_{x_i x_j c}}{N_c + k_{X_i} k_{X_j} - 1}, \quad P^*(c) = \frac{N_c}{N + k_C - 1}, \quad P^*(x_k|c) = \frac{N_{x_k}}{N_c + k_{X_k} - 1}$$

and for $c' \neq c$:

$$P^*(x_i, x_j|c') = \frac{N_{x_i x_j c'} + 1}{N_{c'} + k_{X_i} k_{X_j}}, \quad P^*(c') = \frac{N_{c'} + 1}{N + k_C}, \quad P^*(x_k|c') = \frac{N_{x_k} + 1}{N_{c'} + k_{X_k}}$$

In this way, we assume that the attributes X_i and X_j are suitable to be joined if the following condition is checked:

$$\mathbf{JC}_{L1O}(X_i, X_j) = [JM_{L1O}(X_i, X_j) > 0]$$

Log-Likelihood Ratio Test (LRT). The last approach to decide when joining two variables is based on a log-likelihood ratio test [10]. The log-likelihood ratio test (LRT) have been used to compare two nested models, M_1 and M_2 (in our case M_1 is the model with merged variables and M_2 the simpler model with conditionally independent variables). The log-likelihood ratio criterion is expressed by:

$$LRT = -2. \ln \left(\frac{\sup_\theta P_{M_2}(D|\theta)}{\sup_\theta P_{M_1}(D|\theta)} \right) = -2. \sum_{c,x_i,x_j} N_{cxy} \ln \left(\frac{N_{cx_i}.N_{x_j c}}{N_c.N_{cx_i x_j}} \right)$$

where $P_{M_i}(D|\theta)$ is the likelihood of the data under model M_i and parameters θ. The $\sup_\theta P_{M_i}(D|\theta)$ is obtained computing the likelihood of the data when parameters are estimated with maximum likelihood (in our models the parameters are equal to the relative frequencies in the sample). The third part of the equality shows the closed form to compute LRT.

Asymptotically, LRT is distributed as chi-square random variable with degrees of freedom equal to the difference in the number of parameters between the two models.

In our case, LRT follows a chi-square distribution with $(k_{X_i} - 1)(k_{X_j} - 1)k_C$ degrees of freedom [10], where k_{X_i} is the number of cases of X_i. The null hypothesis (H_0) of the test is X_i and X_j are independent given the class. We consider a significance level α and compute this LRT metric as the p-value of the following test:

$$\mathbf{JM}_{LRT}(X_i, X_j) = \chi^2_{(k_{X_i}-1)k_{X_j}-1)k_C}(LRT)$$

The associated criterion is that the null hypothesis is rejected. But the question is that this test is valid for the comparison of two models, while in our algorithm it is applied many times over the $\frac{n(n-1)}{2}$ possible variable pairs (n is the actual number of active variables) increasing the possibilities of a mistake of the LRT. In order to avoid this effect, the α level is divided by a corrector factor $\rho = \sum_{t=1..R} \frac{1}{t}$ that makes harder the rejection of null hypothesis.

Thus, the joining criterion condition is expressed by:

$$\mathbf{JC}_{LRT}(X_i, X_j) = [JM_{LRT}(X_i, X_j) > (1 - \frac{\alpha}{\rho})]$$

where $R = \frac{n(n-1)}{2}$ is the number of tests, $\mathbf{JM}_{LRT}$ is the joining criterion metric and $\mathbf{JC}_{LRT}$ is the joining criterion condition.

The Joining Algorithm (JA). This algorithm corresponds to the process for joining the dependent variables in a recursive form. It considers the three different joining criteria (i.e. BDe, L10, LRT). The process is quite simple. It joins the variables with highest score, while the joining condition is verified.

Algorithm 1. *Joining Algorithm (**JA**)*

$Z = \{X_1, ..., X_n\}$;
end = false;
while $(\#(Z) \geq 2 \vee \neg end)$
- $\{X_i, X_j\} = arg\max_{\{X_r, X_s\}}\{\boldsymbol{JM}(X_r, X_s) : \{X_r, X_s\} \in Z\}$;
- *if* $\boldsymbol{JC}(X_i, X_j)$ *then*
  - $T = X_i \times X_j$; $Z = Z \setminus \{X_i, X_j\}$; $Z = Z \cup T$;
- *else end=true;*

return Z;

2.2 Grouping Process

As we have already commented, an important problem of joining two attributes is that the number of estimated parameters is much greater, concretely, if X_i and X_j are considered independent given C, $(k_{X_i} + k_{X_j} - 2).k_C$ parameters have to be estimated, while if X_i and X_j are joined, we have to estimate $(k_{X_i}.k_{X_j} - 1).k_C$ parameters. For example, if we join two variables with 7 possible values, the resulting variable will have 49 cases and for some of these combinations, it is possible that there are very few cases in the learning data and the corresponding estimated parameters will have very low reliability.

To solve this problem, we propose a mechanism to group similar states of an attribute. It will be applied to each variable resulting of a joining operation and before any other joining of variables is considered. In this way we try to reduce the number of states before computing the cartesian product with another variable with the expectation of avoiding a combinatorial explosion in the number of states and making easier the possibility of further combinations of this variable.

The process consists in evaluating each pair of states using a given criterion (based on the same principles we used in the joining process), and when the presence of the two states does not suppose a significative benefit with respect to consider them as a unique state, they will be grouped in a single state. Our aim is obviously to reduce the introduced complexity in the model with the joining of the variables.

Grouping Criteria. The same principles used in the Joining Process are valid to define new criteria for this purpose.

To fix the notation, let X be the considered variable. We will assume that x_i and x_j are two possible states of this variable. $X_{\{i,j\}}$ will be the variable in which the elements x_i and x_j has been grouped into a single state. We will work only with the subsample $D_{X(i,j)}$ given by examples from the original sample in which it is verified that $X = x_i$ or $X = x_j$. In order to make the criterion

independent of the other possible states of variable X and their frequencies, we work as if x_i and x_j were the only possible values of variable X. In this situation, groping x_i and x_j implies to build an attribute with a single state. A variable with only one state, is useless. So the grouping criteria will check whether X (with two possible values x_i, x_j) is relevant to C under sample $D_{X(i,j)}$. M_1 will be the more complex model with X relevant to C and M_2 will represent the simpler model in which X is irrelevant to C (x_i and x_j have been grouped).

The metric for grouping x_i and x_j into a single state is denoted by $\mathbf{GM}(x_i, x_j)$ and the condition by $\mathbf{GC}(x_i, x_j)$.

In this way the three evaluated Grouping Criteria are:

BDe Score (BDe). The difference between the BDe scores of making X dependent or independent of C produces:

$$\mathbf{GM}_{BDe}(x_i, x_j) = T_{x_iC} + T_{x_jC} - T_C$$

where

$$T_{x_kC} = \ln\left(\frac{\Gamma(\frac{S}{2})}{\Gamma(\frac{S}{2} + N_{x_k})}\right) \sum_c \ln\left(\frac{\Gamma(\frac{S}{2k_C} + N_{x_kc})}{\Gamma(\frac{S}{2k_C})}\right)$$

$$T_C = \ln\left(\frac{\Gamma(S)}{\Gamma(S + N_{x_i} + N_{x_j})}\right) \sum_c \ln\left(\frac{\Gamma(\frac{S}{k_C} + N_c)}{\Gamma(\frac{S}{k_C})}\right)$$

In these expressions S is a parameter (the global sample size) and the frequencies are measured in subsample $D_{X(i,j)}$.
In this way, we assume that the states x_i and x_j are suitable to be grouped, grouping condition, when:

$$\mathbf{GC}_{BDe}(x_i, x_j) = [GM_{BDe}(x_i, x_j) \leq 0]$$

Leave One-Out Score (L10). As before we compute the logarithm of the likelihood of the learning sample $D_{X(i,j)}$ under the two models M_1 and M_2, but each time we compute the likelihood of a case, we remove this case of the sample to estimate the parameters (this can be done by decreasing the associated frequencies by 1). It is assumed that we apply the Laplace correction to estimate the probabilities. The resulting formula is expressed by:

$$\mathbf{GM}_{L10}(x_i, x_j) = \sum_c N_{x_ic} \left(\ln \frac{N_{x_ic}}{N_{x_i} + k_C - 1}\right) + \sum_c N_{x_jc} \ln\left(\frac{N_{x_jc}}{N_{x_j} + k_C - 1}\right) - \sum_c N_c \ln\left(\frac{N_c}{N_{x_i} + N_{x_j} + (k_C - 1) - 1}\right)$$

The grouping condition is expressed like in BDe metric:

$$\mathbf{GC}_{L10}(x_i, x_j) = [GM_{L10}(x_i, x_j) \leq 0]$$

Log-Likelihood Ratio Test (LRT). As in Section 2.1, we apply the log-likelihood ratio test to compare models M_1 and M_2. The statistic is:

$$LRT = -2.\sum_c N_{x_i c}.\ln\left(\frac{N_{x_i c}(N_{x_i}+N_{x_j})}{N_{x_i}.N_c}\right) - 2.\sum_c N_{x_j c}.\ln\left(\frac{N_{x_j c}(N_{x_i}+N_{x_j})}{N_{x_j}.N_c}\right)$$

The null hypothesis (H_0) is the simpler model M_2 (the model with the states x_i and x_j grouped). In this case LRT follow a chi-square distribution (k_C-1) degree of freedom. The criterion is the p-value of this test:

$$\mathbf{GM}_{LRT}(x_i, x_j) = \chi^2_{k_C-1}(LRT)$$

The associated criterion is that the null hypothesis is rejected. Such as the equivalent joining criterion, the α level is divided by a corrector factor $\rho = \sum_{t=1...R}\frac{1}{t}$ where $R = \frac{k(k-1)}{2}$ is the number of tests and k is the number of active states in the variable X.

$$\mathbf{GC}_{LRT}(x_i, x_j) = [GM_{LRT}(x_i, x_j) > (1 - \frac{\alpha}{\rho})]$$

The Grouping Algorithm (GA). This algorithm is the process for grouping the irrelevant states of a variable in a recursive form. The only variation is the considered grouping criterion. We want to notice the similarity with Algorithm 1. In both cases, a model selection and a model transformation is carried out.

Algorithm 2. *Grouping Algorithm (**GA**)*

$S_X = \{x_1, ..., x_n\}$; *end = false;*
while $(\#(S_X) \geq 2 \vee \neg end)$
- $\{x_i, x_j\} = arg\max_{\{x_r,x_s\}}\{\boldsymbol{GM}(x_r, x_s) : \{x_r, x_s\} \in S_X\}$;
- *if* $\boldsymbol{GC}(x_i, x_j)$ *then*
  - $t = \{x_i \cup x_j\}$; $S_X = S_X \setminus \{x_i, x_j\}$; $S_X = S_X \cup t$;
- *else end=true;*

return S_X;

In the experimental results, presented in Section 3 Table 2, we are going to see that there are not important differences between the three proposed criteria. But the main advantage of the introduction of this grouping process is the reduction in the complexity of final models as well as in the time needed to build the classifier. The experimental results show a reduction of 50% in time.

2.3 The Classifier

Once we have described the joining and grouping processes, we are going to describe how to compose these two processes:

Algorithm 3. *A Semi-naive Bayes with Grouping of Cases (**Semin-NB-G**)*

$Z = \{X_1, ..., X_n\}$;
end = false;
while $(\#(Z) \geq 2 \vee \neg end)$
- $\{X_i, X_j\} = arg\max_{\{X_r, X_s\}}\{JM(X_r, X_s) : \{X_r, X_s\} \in Z\}$;
- *if* $JC(X_i, X_j)$ *then*
  - $T = Grouping(X_i \times X_j)$; $Z = Z \setminus \{X_i, X_j\}$; $Z = Z \cup T$;
- *else end=true;*

return $Grouping(Z)$;

As we can see, the grouping process is applied each time that two attributes to be joined are selected. At the end, the grouping method is applied again over all attributes with the aim of processing the attributes that have not been selected to be joined.

As three distinct metrics can be used in the joining or groping process, there are nine possible schemes to define the classifier. In the next section, we are going to see as the LRT criterion in the joining process and the L1O in the grouping process reach the best results.

The combination of the joining and grouping procedures has an important potential. It can perform some additional preprocessing tasks as side effects as the removing of irrelevant variables.

3 Experimental Results

In this section we are going to carry out an empirical study of the proposed procedures. For this analysis, several data sets have been taken from UCI repository. Due to implementation simplicity reasons, we didn't consider data sets with missing values although our methods could be easily extended. The continuous values are discretized by the Fayyad, Irani method [11]. Due to the lack of space, only the names of the databases are shown: *anneal, balance-scale, german-credit, diabetes, glass, heart-statlog, ionosphere, iris, lymphography, sonar, vehicle, vowel and zoo.*

For the experiments, we have used the Elvira environment [12] and Weka platform [13]. The evaluation of the classifiers was achieved by a 10-fold-cross repeated 10 times scheme and the comparison between classifiers was done using the corrected paired t-test implemented in Weka [13]. The significance level used by all the statistical tests was 0.05.

In that way, we resume the evaluations across the different data sets counting the number of statistically significant improvements (denoted as Wins (**W**) or by the symbol $\circ$), degradations (denoted as Defeats (**D**) or by the symbol $\bullet$) and when there is not any statistically significant change (denoted as Ties (**T**)).

3.1 Evaluating Joining Criteria

For this purpose, we compare the results given by Joining Algorithm (**JA**) using the three proposed criteria without grouping. In this way, we compare these results respect to the naive Bayes classifier, Table 1(a), and respect to the Pazzani's semi-naive Bayes classifier, Table 1(b).

Table 1. Joining Criteria Evaluation

Dataset	NB	JA_{BDe}	JA_{L1O}	JA_{LRT}	NB	JA_{BDe}	JA_{L1O}	JA_{LRT}
anneal	95.95	97.82 ○	96.09	97.29	-0.13	-0.10	-0.14	-0.09 ○
balance-scale	71.56	71.48	69.40 ●	69.40 ●	-0.60	-0.60	-0.54 ○	-0.54 ○
german-credit	75.04	74.54	70.23 ●	73.99	-0.53	-0.53	-0.61 ●	-0.54
pima-diabetes	75.26	75.27	73.31	75.18	-0.54	-0.52 ○	-0.53	-0.51
Glass	71.94	70.32	67.35	71.50	-0.91	-0.90	-1.03	-0.88
heart-statlog	82.56	81.81	75.11 ●	82.89	-0.47	-0.44 ○	-0.49	-0.40 ○
ionosphere	89.40	89.86	65.78 ●	90.09	-1.62	-1.04 ○	-0.50 ○	-0.81 ○
iris	93.33	93.20	91.33	93.33	-0.22	-0.21	-0.29	-0.22
lymphography	85.10	85.75	59.00 ●	86.17	-0.43	-0.40	-0.82 ●	-0.38
sonar	76.71	75.46	55.82 ●	74.98	-0.84	-0.60 ○	-0.67	-0.56 ○
vehicle	61.06	68.63 ○	43.31 ●	68.88 ○	-2.00	-0.68 ○	-1.21 ○	-0.68 ○
vowel	61.99	63.22	62.36	66.57 ○	-1.01	-0.99	-1.83 ●	-0.93 ○
zoo	93.98	92.29	91.39	93.88	-0.12	-0.16	-0.39 ●	-0.12
Average	79.53	79.97	70.81	80.32	-0.73	-0.55	-0.70	-0.51
W/T/D		2/11/0	0/6/7	2/10/1		5/8/0	3/6/4	7/6/0

(%)Percent of cases corrected classified (LL) Mean Log-Likelihood
○, ● statistically significant improvement or degradation

(a) Naive Bayes comparison respect to Joining Algorithm (JA) with the tree proposed joining criteria: BDe, L10 and LRT

	SemiNB	J_{BDe}	J_{L1O}	J_{LRT}
Average	78.83	79.97	70.81	80.32
W/T/D		0/12/1	0/6/7	0/12/1

(%)Percent of cases corrected classified

SemiNB	J_{BDe}	J_{L1O}	J_{LRT}
-0.58	-0.55	-0.70	-0.51
	4/8/1	1/6/6	4/8/1

(LL) Mean Log-Likelihood

(b) Semi-NB comparison respect to Joining Algorithm (JA) with the tree proposed joining criteria: BDe, L10 and LRT

As we can see in Table 1(a), the BDe and LRT criteria allow to outperform the naive Bayes classifier in terms of accuracy and mean log-likelihood. In fact, the joining process is designed to minimize the log-likelihood metric and it achieves it in a very robust way, in none of the once data bases there is a significant deterioration. In addition, these two criteria reach the performance of the semi-naive Bayes classifier [1], Table 1(b). LTR criterion appears to be slightly better than BDe. On the other side, the L10 performance is clearly bad. We have not found any founded reason for that. One possible reason is L10 criterion tests the joining of two variables without considering the other ones. The experiments are carried out with the full set of attributes and then the final behaviour can be different from what was initially expected.

3.2 Evaluating Grouping Criteria

In previous subsection, the LRT criterion has been selected as the most suitable one for joining variables. So we fix it and now we are going to evaluate which criterion could be the most suitable for grouping cases using the Semin-naive Bayes with Grouping of Cases (**SNB-G**). In the Table 2, the obtained results with the three possible criteria are shown. As we can see, there is almost no difference among them, though L1O criterion seems to be the best one. Here it does not suffer the same problem when it was applied to joining variables. Perhaps, this is due to the fact that grouping is simpler operator than joining.

Table 2. Grouping Criteria Evaluation

Dataset	NB	G_{BDe}	G_{L1O}	G_{LRT}	NB	G_{BDe}	G_{L1O}	G_{LRT}
Average	79.53	80.26	80.61	79.75	-0.73	-0.49	-0.49	-0.50
W/T/D		2/11/0	3/9/0	2/11/0		6/7/0	7/6/0	5/8/0

(%)Percent of cases corrected classified (LL) Mean Log-Likelihood

Description: NB comparison respect to SNB-G with the J_{LRT} joining criterion and the three proposed Grouping Criteria: BDe, L10 and LRT.

3.3 Comparing with Pazzani Semi-naive Bayes

In Table 1(b) we could see as the LRT is a good metric for joining variables, in fact, it reaches the performance of the semi-naive Bayes with a less computational cost. In this subsection, we are going to compare the SNB-G algorithm, with a joining process using the LRT metric and a grouping of cases using the L1O metric. These results are show in Table 3(a)(b) and the model building time comparison is shown in Table 3(c).

Table 3. Semi-NB, **SNB-G** and NB Comparison

Dataset	Semi-NB	**SNB-G**	NB
Average	78.83	80.61	79.53
W/T/D		1/12/0	1/10/2

(a) Percent of corrected classified

Semi-NB	**SNB-G**	NB
-0.58	-0.49	-0.73
	5/7/1	4/6/3

(b) Mean Log-Likelihood

Semi-NB	**SNB-G**
40.34	0.54

(c) Building Time (sec)

In the Table 3, we can see as our proposed procedure reaches the performance of the Panzzani's semi-naive Bayes classifier in terms of percentage of correct classifications while it outperformances it in terms of log-likelihood. In addition to this, we can see as our algorithm is more robust than the Pazzani's classifier when it is compared respect to the naive Bayes classifier because the performance of Pazzani's classifier is worse in a significative way in several data bases.

Another main point to analyze is the compared computational cost of these two approaches. In Table 3(c), we can see as the average model building time of the Pazzani's semi-naive Bayes is eighty times greater than our approach.

3.4 State of the Art Classifiers Comparison

In this section, we compare our approach with the state of the art of the classifiers with the best performance in the UCI repository. These classifiers are naive Bayes (NB), Tree-Augmented Naive Bayes (TAN), Averaged One-Dependence Estimators (AODE) [15] and Lazy Bayesian Rules (LBR) [15]. Also it is included the result of the semi-naive Bayesian classifier of Pazzani [1] again.

As we can see in Table 4, our proposal SNB-G outperforms the Naive Bayes classifier in terms of accuracy prediction and mean of log-likelihood in a robust way. At the same time, its performance is comparable to AODE, TAN and LBR and it outperforms the Pazzani's Semi-NB.

Table 4. SNB-G state of the art classifiers Comparison

Dataset	**SNB-G**	NB	AODE	TAN	LBR	Semi-NB
Average	80.61	79.53	81.57	81.53	81.10	78.83
W/T/D		0/10/3	1/11/1	0/13/0	0/13/0	0/12/1

(%)Percent of cases corrected classified

SNB-G	NB	AODE	TAN	LBR	Semi-NB
-0.49	-0.73	-0.48	-0.49	-0.60	-0.58
	0/6/7	4/8/1	0/12/1	0/8/5	1/7/5

(LL) Mean Log-Likelihood

4 Conclusion and Future Work

In this paper we have presented a combination of two procedures as a preprocessing step for a naive Bayes classifier: a method to join variables and a method to group cases of a variables. We have shown that the combined application of them is very efficient and the performance is similar to more costly wrapper methods in terms of accuracy and better in terms of log-likelihood.

The main points for future researh are the application to real problems in which there are a high number of variables and fast classifiers are convenient (for example the junk mail classification problem) and also the generalization of the methodology to other models, as classification trees or TAN models.

References

1. Pazzani, M.J.: Searching for dependencies in bayesian classifiers. Lecture Notes in Statistics, vol. 112, pp. 239–248 (1995)
2. Abellán, J.: Application of Uncertainty Measures on Credal Sets on the Naive Bayes Classifier. Int. J. of General Systems 35(6), 675–686 (2006)
3. Abellán, J., Moral, S.: Upper entropy of credal sets. Applications to credal classification. Int. J. of Approximate Reasoning 39(2-3), 235–255 (2005)
4. Abellán, J., Klir, G.J., Moral, S.: Disaggregated total uncertainty measure for credal sets. Int. J. of General Systems 35(1), 29–44 (2006)
5. Kononenko, I.: Semi-naive bayesian classifier. In: EWSL-91: Proceedings of the European working session on learning on Machine learning, pp. 206–219. Springer, Heidelberg (1991)
6. Zheng, F., Webb, G.: A comparative study of semi-naive bayes methods in classification learning. In: Proc. 4th Australasian Data Mining conference, pp. 141–156 (2005)
7. Domingos, P., Pazzani, M.J.: On the optimality of the simple bayesian classifier under zero-one loss. Machine Learning 29(2-3), 103–130 (1997)
8. Heckerman, D., Geiger, D., Chickering, D.M.: Learning bayesian networks: The combination of knowledge and statistical data. In: KDD Workshop, pp. 85–96 (1994)
9. Neapolitan, R.E.: Learning Bayesian Networks. Prentice-Hall, Englewood Cliffs (2004)
10. Wilks, S.S.: The large-sample distribution of the likelihood ratio for testing composite hypotheses. Annals of Mathematical Statistics 9, 60–62 (1938)
11. Fayyad, U., Irani, K.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proc. of 13th Int. Joint Conf. on AI (1993)
12. Consortium, E.: Elvira: An environment for probabilistic graphical models. In: Proceedings of the 1st European Workshop on Probabilistic Graphical Models (2002)
13. Witten, I.H., Frank, E.: Data mining: practical machine learning tools and techniques with Java implementations. Morgan Kaufmann, San Francisco (2000)
14. Duda: Pattern Classification and Scene Analysis. John Wiley Sons, Chichester (1973)
15. Webb, G.I., Boughton, J.R., Wang, Z.: Not so naive bayes: Aggregating one-dependence estimators. Mach. Learn. 58(1), 5–24 (2005)

Split Criterions for Variable Selection Using Decision Trees*

Joaquín Abellán and Andrés R. Masegosa

Department of Computer Science and Artificial Intelligence.
University of Granada, Spain
{jabellan,andrew}@decsai.ugr.es

Abstract. In the field of attribute mining, several feature selection methods have recently appeared indicating that the use of sets of decision trees learnt from a data set can be an useful tool for selecting relevant and informative variables regarding to a main class variable. With this aim, in this study, we claim that the use of a new split criterion to build decision trees outperforms another classic split criterions for variable selection purposes. We present an experimental study on a wide and different set of databases using only one decision tree with each split criterion to select variables for the Naive Bayes classifier.

Keywords: Variable selection, classification, decision tree, Naive Bayes, Imprecise probabilities, uncertainty measures.

1 Introduction

It is possible to apply classification methods on a given database containing several samples where each sample also contains a set of values belonging to an attribute or predictive variable set and a variable labeled *class variable*. In the field of machine learning, the classification subject is based on the use of several techniques that infer rules from a given data set so as to be able to predict new values of the class variable (discrete or discretized) using a new set of values for the remaining variables (known as *attribute variables*). The data set used to obtain these rules is labeled the training data set and the data set used to check the classifier is called the test data set. Classification has important and distinguished applications in the fields of medicine, bioinformatics, physics, pattern recognition, economics, etc., and is used for disease diagnosis, meteorological forecasts, insurance, text classification, etc.

The existence of redundant or irrelevant variables in a database can cause a deterioration in the performance of many classification methods. When databases contain a large number of variables, as in bioinformatics or text field classification where there are commonly several thousands of variables (genes or words),

* This work has been supported by the Spanish Ministry of Science and Technology under the projects TIN2005-02516 and TIN2004-06204-C03-02 and FPU scholarship programme (AP2004-4678).

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 489–500, 2007.

this excessive amount of information could become unmanageable. It is therefore necessary to select a smaller number of variables in order to reduce or remove any irrelevant or redundant information present in the data and to enable automatic treatment of these data sets. In order to solve this topic, several methods have appeared which obtain a significant subset of variables for classification purposes. In Hall and Holmes [11], we can find a detailed description of this kind of method.

In recent literature, several methods have been defined which use decision trees to select variables with very good results. Li et al. [14], for example, detail a selection variable method for genetic databases that uses 20 decision trees learnt from the training data set. In Lau and Schultz [13], a large number of simple decision trees built from several partitions of the data set are used with the aim of selecting relevant genes. In Ratanamahatana and Gunopulos [18], five decision trees are learnt from small partitions of the training data set to obtain the variable subset. What all these methods have in common is that they use the first levels of the decision trees where the most significant variables are.

In this article, we will use the Naive Bayes classifier [9] as a base classification method and we will use a filter[1] method for variable selection that builds decision trees independently of the Naive Bayes classifier. In fact, this way of selecting variables using a decision tree should not favour the Naive Bayes classifier because this is based on the independency of the attribute variables given the class variable and when a decision tree is used for variable selection, correlated variables are obtained, also a decision tree could not delete redundant variables. But, our target here is not to present a variable selection method using only one decision tree, we want to compare split criterions for this aim. In the future, we want to present complex methods of variable selection that use a set of decision trees obtained with the best split criterion possible.

In this paper, the selected variables are taken from the first levels of the decision trees that will be built using four different split criterions. Three of them have been widely used: Info-Gain, used in Quinlan's ID3 classifier [16]; Info-Gain Ratio, used in Quinlan's C4.5 classifier [17]; Gini Index, widely used in statistics as an impurity measure of a partition. We are going to compare these three split criterions with a previously proposed one, the Imprecise Info-Gain [3], also successful used to build classification decision trees. This last measure is based on a maximum entropy measure on the imprecise Dirichlet model [21].

We will see how by using the set of variables in the first three and the first four levels of one decision tree we can obtain a smaller variable subset in relation to the original one. In addition, by using the Naive Bayes classifier with this smaller variable subset, we will attain the same performance as if the whole variable set was used. Although (as we have already mentioned) the aim of this work is not to propose a variable selection method using a single decision tree, we intend to show how the decision trees built with the Imprecise Info-Gain split criterion get smaller and more informative feature subsets than the classic split criterions. In that way, this split criterion is likely to be better when it would be used in the

[1] It not depend on the classification method used.

already mentioned complex variable selection methods based on many decision trees, which should provide a more informative attribute variable subset than the obtained one with a single decision tree.

For this analysis, we will use a wide series of databases which are commonly used in classification tasks. Using these databases, the performance of the Naive Bayes will be analyzed before and after the application of each variable selection method and also the number of their selected variables. The computational cost of these variable selection methods will also be evaluated using another special series of data sets with a large number of variables.

Section 2 briefly describes the Naive Bayes classifier (which is used as a base classifier) and also describes the set of split criteria used for building the decision trees that are simultaneously used in the variable selection scheme. Section 3 shows the databases used in the experimental work as well as the obtained results. Section 4 analyzes the experimental results. Finally, Section 5 explores the conclusions and future lines of work.

2 Previous Knowledge

2.1 Naive Bayes

The success of the model developed by Duda and Hart [9] is mainly due to its simplicity, efficiency and effectiveness in classification problems. Before describing the classifier, we will probabilistically describe the supervised classification problem.

Let $\mathcal{D}$ be a database, with size N, and with values in a set $\mathcal{L}$ of (discrete or discretized) attribute variables $\{X_i | \, i = 1, \ldots, r\}$, where each variable has a set of possible states or cases $\Omega_{X_i} = \{x_1^i, x_2^i, ..., x_{|\Omega_{X_i}|}^i\}$, and a class variable C, whose states are $\Omega_C = \{c_1, c_2, ..., c_k\}$. The objective is to obtain information from the database so that given an observation (a set of values of all the attribute variables) it is possible to associate this with a value of the class variable.

If we represent the new sample as $\mathbf{x}$, with $\mathbf{x} = \{x_{h_i}^1, .., x_{h_r}^r\}$. The Naive Bayes predicts value c_i in the following way: $arg \max_{c_i} \left(P(c_i|\mathbf{x})\right)$.

Now, based on the supposition that the attribute variables are independent given the class variable, it can be expressed as

$$arg \max_{c_i} \left(P(c_i) \prod_{j=1}^{r} P(x_{h_j}^j | c_i) \right).$$

2.2 Decision Trees and Split Criteria

A decision tree is a simple structure that can be used as a classifier. Within a decision tree, each node represents an attribute variable and each branch represents one of the states of this variable. A tree leaf specifies the expected value of the class variable depending on the information contained in the training data set. When we obtain a new sample or instance of the test data set, we can make

a decision or prediction about the state of the class variable following the path to the tree from the root until a leaf using the sample values and the tree structure. Associated to each node is the most informative variable which has not already been selected in the path from the root to this node (as long as this variable provides more information than if it had not been included). In this last case, a leaf node is added with the most probable class value for the partition of the data set defined with the configuration given by the path until the tree root.

In order to measure the quantity of information, several criteria or metrics can be used, and these are called split criteria. In this article, we will analyze the following ones: Info-Gain, Info-Gain Ratio, Gini Index, and Imprecise Info-Gain.

Info-Gain [IG]. This metric was introduced by Quinlan as the basis for his ID3 model [16]. This model has the following main features: it was defined to obtain decision trees with discrete variables, it does not work with missing values, a pruning process is not carried out, and it is based on Shannon's entropy [19]. This split criterion can therefore be defined on a variable X_i given the class variable C in the following way:

$$IG(X_i, C) = H(C) - H(C|X_i),$$

where $H(C)$ is the entropy of C: $H(C) = -\sum_j p(c_j) \log p(c_j)$, with $p(c_j) = p(C = c_j)$, the probability of each value of the class variable estimated in the training data set; and $H(X_i)$ is known as *split info*. In the same way, $H(C|X_i) = -\sum_t \sum_j p(c_j|x_t^i) \log p(c_j|x_t^i)$. Finally, we can obtain the following reduced expression for the Info-Gain criterion:

$$IG(X_i, C) = -\sum_t \sum_j p(c_j, x_t^i) \log \frac{p(c_j, x_t^i)}{p(c_j)p(x_t^i)}.$$

This criterion is also known as the *Mutual Information Criterion* and is widely used for measuring the dependence degree between an attribute variable and the class variable. It tends to select attribute variables with many states and consequently results in excessive ramification.

Info-Gain Ratio [IGR]. In order to improve the ID3 model, Quinlan introduces the C4.5 model, where the Info-Gain split criterion is replaced by an Info-Gain Ratio criterion which penalizes variables with many states. A procedure can then be defined to work with continuous variables, it is possible to work with missing data, and a posterior pruning process is introduced.

The Info-Gain Ratio of an attribute variable X_i on a class variable C can be expressed as:

$$IGR(X_i, C) = \frac{IG(X_i, C)}{H(X_i)}.$$

Gini Index [GIx]. This criterion is widely used in statistics for measuring the impurity degree of a partition of a data set in relation to a given class variable

(we can say that a partition is "pure" when it only has a single associated value of the class variable). The work by Breiman et al. [8] can be mentioned as a reference for the use of the Gini Index in decision trees.

In a given database, the Gini Index of a variable X_i can be defined as:

$$gini(X_i) = \sum_j (1 - p^2(x_j^i)).$$

In this way, we can define the split criterion based on the Gini Index as:
$GIx(X_i, C) = gini(C|X_i) - gini(C)$,
where

$$gini(C|X_i) = \sum_t p(x_t^i) gini(C|X_i = x_t^i).$$

We can see that the expression GIx is written in a different way to that used for the previous split criteria because now the variable with the highest $gini(C|X_i)$ value is selected (contrary to what happens with the entropy).

Imprecise Info-Gain [IIG]. The Imprecise Info-Gain criterion was first used for building simple decision trees in Abellán and Moral's method [3] and in a more complex procedure in Abellán and Moral [5]. In a similar way to ID3, this tree is only defined for discrete variables, it cannot work with missing values, and it does not carry out a posterior pruning process. It is based on the application of uncertainty measures on convex sets of probability distributions. More specifically, probability intervals are extracted from the database for each case of the class variable using Walley's imprecise Dirichlet model (IDM) [21], which represents a specific kind of convex sets of probability distributions, and on these the entropy maximum is estimated. This is a total measure which is well known for this type of set (see Abellán, Klir and Moral [6]).

The IDM depends on a hyperparameter s and it estimates that (in a given database) the probabilities for each value of the class variable are within the interval:

$$p(c_j) \in \left[\frac{n_{c_j}}{N+s}, \frac{n_{c_j}+s}{N+s} \right],$$

with n_{c_j} as the frequency of the set of values $(C = c_j)$ in the database. The value of parameter s determines the speed with which the upper and lower probability values converge when the sample size increases. Higher values of s give a more cautious inference. Walley [21] does not give a definitive recommendation for the value of this parameter but he suggests values between $s = 1$ and $s = 2$. In Bernard [7], we can find reasons in favor of values greater than 1 for s.

If we label $K(C)$ and $K(C|(X_i = x_t^i))$ for the following sets of probability distributions q on Ω_C:

$$K(C) = \left\{ q | \, q(c_j) \in \left[\frac{n_{c_j}}{N+s}, \frac{n_{c_j}+s}{N+s} \right] \right\},$$

$$K(C|(X_i = x_t^i)) = \left\{ q | \, q(c_j) \in \left[\frac{n_{\{c_j, x_t^i\}}}{N+s}, \frac{n_{\{c_j, x_t^i\}}+s}{N+s} \right] \right\},$$

with $n_{\{c_j,x_t^i\}}$ as the frequency of the set of values $\{C = c_j, X_i = x_t^i\}$ in the database, we can define the Imprecise Info-Gain for each variable X_i as:

$$\mathbf{IIG}(X_i, C) = S(K(C)) - \sum_t p(x_t^i) S(K(C|(X_i = x_t^i))),$$

where $S()$ is the maximum entropy function of a convex set.

For the previously defined intervals and for a value of s between 1 and 2, it is very easy to obtain the maximum entropy using procedures of Abellán and Moral [2,4] or the specific one for the IDM of Abellán [1].

3 Experimentation

In order to experimentally validate how the Imprecise Info-Gain (**IIG**) split criterion is more informative than the classic ones, we have built a decision tree for each data set with each of the exposed criteria in the previous section. We have selected the variables that independently appear in the first three or the first four levels of these trees and we have applied the Naive Bayes classifier before and after the selection. For the **IIG** criterion, we have used a value of $s = 1.5$ because it obtains a trade-off between size of the tree and classification accuracy (considering all the values between $s = 1$ and $s = 2$).

In order to check the above procedure, we have used a wide and different set of 27 known databases, obtained from the *UCI repository of machine learning databases* which can be directly downloaded from ftp://ftp.ics.uci.edu/machine-learning-databases. A brief description of these can be found in Table 1, where column "N" is the number of instances in the databases, column "Att" is the number of attribute variables, column "Nom" is the number of nominal variables, column "k" is the number of cases or states of the class variable (always a nominal variable) and column "Rang" is the range of states of the nominal variables of each database.

Table 1. Data Base Description

Data Base	N	Att	Num	Nom	k	Range
1. Anneal	898	38	6	32	6	2-10
2. Audiology	226	69	0	69	24	2-6
3. Autos	205	25	15	10	7	2-22
4. Br-cancer	286	9	0	9	2	2-13
5. Colic	368	22	7	15	2	2-6
6. Cr-german	1000	20	7	13	2	2-11
7. Diab-pima	768	8	8	0	2	-
8. Glass-2	163	9	9	0	2	-
9. Hepatitis	155	19	4	15	2	2
10. Hypothyroid	3772	29	7	22	4	2-4
11. Ionosfere	351	35	35	0	2	-
12. Kr-vs-kp	3196	36	0	36	2	2-3
13. Labor	57	16	8	8	2	2-3
14. Letter	20000	16	16	0	2	-
15 Lymph	146	18	3	15	4	2-8
. . . . .						
16. Mfeat-pix	2000	240	240	0	10	5-7
17. Mushroom	8123	22	0	22	2	2-12
18. Optdigits	5620	64	64	0	10	-
19. Segment	2310	19	16	0	7	-
20. Sick	3772	29	7	22	2	2
21. Sol-flare1	323	12	0	12	2	2-6
22. Sonar	208	60	60	0	2	-
23. Soybean	683	35	0	35	19	2-7
24. Sponge	76	44	0	44	3	2-9
25. Vote	435	16	0	16	2	2
26. Vowel	990	11	10	1	11	2
27. Zoo	101	16	1	16	7	2
Ad	3279	1558	1555	3	2	2
Isolet	7797	617	617	0	26	-
Ovarian	253	15154	15154	0	2	-

For our experimentation, we have used *Weka* software [22] on Java 1.5, and we have added the necessary methods to build decision trees using the **IIG** split criterion and for the others split criteria. Also, we have added the filters needed to automatically eliminate the variables which are not selected by any of the split criteria.

We have applied the following preprocessing: databases with missing values have been replaced with mean values (for continuous variables) and mode (for discrete variables) using *Weka's* own filters. In the same way, continuous variables have been discretized in five equal frequency intervals. Using equal frequency discretization is therefore of no benefit to any of the selection methods presented, as with Fayyad and Irani's discretization method [10] based on entropy.

In the experimentation, for each preprocessed database we have applied the following procedure of 10-fold-cross validation repeated 10 times: for each partition of training/test sets we build a decision tree using the training set and select the variables in the first three levels of the tree; we select this set of variables on the original database, i.e. without preprocessing, with the same partitions and apply Naive Bayes on these training/test sets generated. The procedure is repeated using now the first fourth levels. These results are compared following the same validation procedure using the results of the Naive Bayes classifier (with no variable selection scheme).

In Table 2 we can see the results of the right classification percentage using the variables of the first three levels of the decision tree for each split criterion and the Naive Bayes as the base classifier. Column NB contains the results of the Naive Bayes Classier using the whole set of variables of the data set and the others the results for the selection scheme using the different split criteria. This table also presents a test for statistically significant differences for the accuracy means between the Naive Bayes and for each of the proposed selection schemes. This test is a corrected two-tailed paired t-test [15] with a 5% significance level[2]. $\circ$ represents a statistically significant improvement in the accuracy mean in relation to the Naive Bayes and also $\bullet$ a statistically significant deterioration in the accuracy mean in relation to the Naive Bayes. In the last row of this table (W/T/D), we can see the accumulated number of wins ($\circ$), ties and defeats ($\bullet$) in the statistical tests for each method and in relation to the Naive Bayes. Table 4 shows the mean number of selected variables using the same notation and the same procedure to compare this measure.

If we fix one of the split criterions as base classifier and we compare with it the another three criterions, a total of 81 ($= 3$ criterions $\cdot$ 27 databases) comparisons can be achieved. All the wins and defeats (using the already mentioned T-test at 5%) in these 81 comparisons for each split criterion are shown in Table 3(a) as percentage values. In Table 3(b), it is shown the distribution of wins, ties and defeats in the number of selected variables when there is not any statistically significative difference in the accuracy prediction between the split criterions.

[2] These test have been widely presented in the literature for this type of experiments, for example in Hall and Holmes [11]. As reviewers comment us, consider others non-parametric test for this type of experiments could be a better choice.

Table 2. Right Classifications Percentage using 3 levels

DB	NB	IG	IGR	GIx	**IIG**
1.	86.59	85.24 •	82.60 •	82.32 •	83.63 •
2.	72.64	73.14	27.95 •	70.86	71.76
3.	57.41	60.13	59.70	59.79	61.87
4.	72.70	72.35	72.56	72.45	73.83
5.	78.70	81.99 ○	84.33 ○	83.26 ○	83.56 ○
6.	75.16	74.79	73.94	74.70	73.61
7.	75.75	75.89	75.90	75.77	76.07
8.	62.43	62.01	62.07	61.88	62.33
9.	83.81	82.37	81.62	82.56	81.18
10.	95.30	94.63 •	94.34 •	94.34 •	94.19 •
11.	82.17	89.43 ○	87.47 ○	87.98 ○	84.34
12.	87.79	90.43 ○	90.43 ○	90.43 ○	90.43 ○
13.	93.57	87.40	83.50	88.77	88.97
14.	64.07	63.93	62.40 •	65.28 ○	63.93
15.	83.13	81.47	79.37	80.87	77.81
. . .					
16.	93.36	85.10 •	83.77 •	86.85 •	89.26 •
17.	95.76	98.85 ○	98.76 ○	98.80 ○	98.44 ○
18.	91.39	88.36 •	29.53 •	88.29 •	88.59 •
19.	80.17	81.41	80.27	80.36	87.37 ○
20.	92.75	93.97 ○	95.26 ○	94.06 ○	96.33 ○
21.	93.02	95.58 ○	96.08 ○	95.92 ○	97.31 ○
22.	67.71	63.44	67.41	63.82	67.58
23.	92.94	84.44 •	35.56 •	78.11 •	86.11 •
24.	92.11	92.82	94.75	93.45	95.00
25.	90.02	94.21 ○	93.58 ○	93.56 ○	94.92 ○
26.	66.79	66.07	64.66	65.77	65.31
27.	95.07	92.98	83.05 •	92.69	92.89
Aver	82.31	81.94	75.59	81.59	82.47
W/T/D		7/15/5	7/12/8	8/14/5	7/15/5

○, • statistically significant improvement or degradation

Table 3. Split Criterion Comparison with 3 levels

Method	W(%)	D (%)
IG	12.3	6.2
IGR	2.5	24.7
GIx	9.9	8.6
IIG	18.5	3.7

(a) Wins and Defeats percentage over the prediction accuracy

W_{NV}(%)	D_{NV}(%)	T_{NV}(%)
13.6	28.4	39.5
21.0	17.3	34.6
2.5	40.7	38.3
54.3	4.9	18.5

(b) W, D and T percentage over the number of variables when the accuracy is the same

Total W(%)	*Total D*(%)
25.9	34.6
23.5	42.0
12.3	49.4
72.8	8.6

(c) Accumulated percentage of Wins and Defeats in prediction accuracy and number of variables

In this, one wins indicates a statistically significant smaller number of selected variables.

In that way, IG criterion gets a better prediction accuracy in the 12.3% of the cases and a worst prediction accuracy in the 6.2% of the comparisons. When the prediction accuracy is the same, IG selects a smaller number of variables in the 13.6% of the total comparisons; a greater number in the 28.4% and the same number in the 39.5% (Note that 12.3%+6.2%+13.6%+28.4%+39.5%=100%). In that way, IG outperforms the another three criterions in the 25.9% (= 12.3% + 13.6%) of the times, while IG is outperformed by one criterion in the 34.6% (= 6.2% + 28.4%) of the comparisons.

In Tables 5 and 6, we can see the same analysis and comparisons as in the previous cases but using the first four levels of the decision trees for variable selection.

In order to analyze the efficiency and computing time of each criterion, we have built a decision tree with only the first three levels for each of the following databases with a large number of variables: *Ad*, *Isolet* and *Ovarian*. The first two databases have been discretized with Fayad and Irani's method [10] as there is hardly any variable with one state (only one in *Isolet*); *Ovarian* has been discretized with the equal frequency method for the same reasons as in the the previous 27 databases. The results for the tree size and the tree-building

Table 4. Mean Number of Selected Variables using 3 levels

DB	NB	IG	IGR	GIx	**IIG**
1.	38.00	8.90 ○	3.00 ○	5.68 ○	7.09 ○
2.	69.00	6.71 ○	3.67 ○	6.70 ○	4.94 ○
3.	25.00	11.27 ○	10.79 ○	11.89 ○	9.76 ○
4.	9.00	7.79 ○	6.90 ○	7.77 ○	3.73 ○
5.	22.00	8.02 ○	8.04 ○	7.91 ○	5.28 ○
6.	20.00	11.65 ○	10.56 ○	11.27 ○	9.89 ○
7.	8.00	7.92	7.90	7.94	3.20 ○
8.	9.00	7.52 ○	8.00 ○	7.75 ○	5.39 ○
9.	19.00	8.38 ○	5.52 ○	8.25 ○	5.68 ○
10.	29.00	6.60 ○	6.60 ○	6.60 ○	3.00 ○
11.	34.00	12.03 ○	6.51 ○	13.01 ○	4.78 ○
12.	36.00	3.00 ○	3.00 ○	3.90 ○	3.00 ○
13.	16.00	5.05 ○	3.84 ○	5.01 ○	2.87 ○
14.	16.00	8.00 ○	9.02 ○	9.26 ○	8.00 ○
15.	18.00	8.68 ○	6.10 ○	8.49 ○	6.64 ○
. . .					
16.	240.00	30.94 ○	23.87 ○	32.21 ○	30.21 ○
17.	22.00	3.00 ○	3.05 ○	3.00 ○	3.00 ○
18.	64.00	18.51 ○	5.14 ○	20.32 ○	19.03 ○
19.	19.00	9.63 ○	11.76 ○	11.12 ○	10.68 ○
20.	29.00	7.30 ○	7.27 ○	8.14 ○	4.82 ○
21.	12.00	6.59 ○	6.65 ○	6.41 ○	0.59 ○
22.	60.00	15.29 ○	17.69 ○	17.32 ○	7.82 ○
23.	35.00	12.37 ○	3.00 ○	9.92 ○	12.07 ○
24.	44.00	3.66 ○	3.00 ○	3.66 ○	2.07 ○
25.	16.00	4.77 ○	5.47 ○	5.19 ○	3.33 ○
26.	11.00	9.45 ○	9.57 ○	9.01 ○	9.72 ○
27.	16.00	5.99 ○	3.60 ○	6.02 ○	6.41 ○
Aver	34.67	9.22	7.39	9.40	7.15
W/T/D		26/1/0	26/1/0	26/1/0	27/0/0

○, ● statistically significant improvement or degradation

Table 5. Split Criterions Comparison respect to NB using 4 levels

	NB	IG	IGR	GIx	**IIG**
Mean	82.31	82.41	78.87	82.28	82.96
W/T/D		7/16/4	9/11/7	8/15/4	8/15/4

(a) Mean of the right classification percentage across data bases and num. of Wins(W), Ties (T) and Defeats(D) comparison respect to NB.

Original	IG	IGR	GIx	**IIG**
34.67	12.80	11.01	13.10	10.91
	0/3/24	0/2/25	0/3/24	0/1/26

(b) Mean number of selected variables across data bases and num. of W, T and D comparison respect to original number of variables.

Table 6. Split Criterions Comparison with 4 levels

Method	W(%)	D (%)
IG	6.2	11.1
IGR	4.9	13.6
GIx	4.9	7.4
IIG	17.3	1.2

(a) Wins and Defeats percent. over the prediction accuracy

$W_{NV}(\%)$	$D_{NV}(\%)$	$T_{NV}(\%)$
13.6	23.5	45.7
16.0	28.4	37.0
8.6	34.6	44.4
55.6	7.4	18.5

(b) W, D and T percentage over the number of variables when the accuracy is the same

Total $W(\%)$	*Total* $D(\%)$
19.8	34.6
21.0	42.0
13.6	42.0
72.8	8.6

(c) Accumulated percentage of W and D in prediction accuracy and num. of vars.

Table 7. Tree Size/Computing Time (seconds) for 3 levels using the whole set

Data Base	IG	IGR	GIx	**IIG**
Ad	126/6.36	13/6.53	146/6.3	58/6.55
Isolet	353/7.78	55/9.38	369/7.86	244/11.2
Ovarian	26/3.75	19/3.41	26/3.41	19/3.34
Mean (nodes/sec)	24.04	4.48	25.91	12.11

processing time can be seen in Table 7. In the final row, we can also see the mean number of nodes that each method can create per second in these databases.

4 Result Analysis

We will first analyze the results jointly. Looking at Table 2, we can see that although variables from only one tree are selected, each criterion performs well (except for some databases with the IGR criterion) in relation to the Naive Bayes while using a very small number of variables (see Table 4). We can therefore see how these are both promising and efficient methods for selecting variables although (as we mentioned at the start) the aim of this work is not to propose a selection variable method but rather to analyze which of these criteria would be the most suitable for use in the complex variable selection methods based on the building of a large number of decision trees.

The obtained improvement using four levels (Table 5) is generally not as significant if we take into account that 20% and 30% more variables are selected. In fact, the most significant variables in relation to the class variable are usually in the first levels of the decision trees. In return, with the IGR criterion, the obtained improvement in the change between the first three and the first four levels is more significant; we must take into account that this method usually selects variables with a lower number of states and therefore needs more levels of the tree to reach a high number of variables.

We see in Tables 2 and 5 that there are not many differences between the results obtained using any of the criteria for selecting variables and the results obtained by the Naive Bayes, if we see the number of statistical tests W/T/D shown in the last row. The IGR criterion is notorious for obtaining bad results in certain databases such as *Audiology, Optdigits* and *Soybean* where the penalty of this method over the variables with a high number of states might be excessive.

In order to compare the results in relation to the accuracy between the different methods, firstly, we must pay attention to Tables 3(a) and Table 6(a). In the first table, we can see how the **IIG** method is statistically better than the others. It outperforms in the 18.5% and only is outperformed by a 3.7% of the cases. For four levels is a bit better.

Another strong point in favour of IIG appears in Tables 3(b) and 6(b). When the split criterions get the same accuracy, IIG selects a statistically significative smaller number of variables in the 54.3% of the cases with three levels and the 55.6% with four levels.

In that way, Tables 3(c) and 6(c) show the percentage of comparisons where there is an improvement in the prediction accuracy or in the number of selected variables when there is the same accuracy. As we can see, IIG gets an outperforming in the 72.8% of the cases and it is only defeated in the 8.6%. That performing is much better than whichever of the another three split criterions.

As we can see, **IIG** is a definitive winner in this study considering a trade-off of the results and the number of variables obtained using first levels of a single decision tree. IIG is the only one that improves on the mean accuracy of the Naive Bayes using the first three and the first four levels. In absolute terms, with 3 levels, NB is outperformed by IIG in 12 cases and NB outperform IIG in 15 cases. With 4 levels, the contrary situations happens and if we consider all the variables obtained by a tree with the IIG criterion, NB is outperformed

in 15 cases and outperform IIG in 11 cases, being IIG again the best criterion considering accuracy and number of variables (we can not present all these results here by limitations of space for this paper).

About the others criteria, we can see that IG and GIx are very similar with respect to the percentage of right classifications and the number of selected variables, with the IGR performing worst in relation to the mean accuracy although better than IG and GIx in terms of the number of variables.

In order to apply these simple methods as the basis for other more complex ones, we must take into account the time needed for each method to build the decision tree. For this purpose, we can analyze the study carried out with the databases with a high number of attribute variables in Table 7. It is evident that both IG and GIx can build more nodes in the decision tree per second, followed by **IIG** and then IGR (which takes on average three times as long as **IIG**). However, if we take into account the time spent building the tree, we can see that there is hardly any difference between them , except in *Isolet* where **IIG** obtains the worst results.

Using the times in Table 7, in our opinion, none of the methods stands out as negative. Perhaps the only bad data for the **IIG** criterion is that the *Isolet* tree was the most costly in terms of the computational time, and **IIG** spends 20% longer than IGR and 42% longer than IG or GIx, but data about the average number of nodes built per second and the time needed in the other two databases with an extremely large number of variables indicates that these differences are not very important.

5 Conclusions and Future Work

Using the known Naive Bayes classifier as a reference, we have introduced an experimental study of split criterions for variable selection methods on this classifier. These methods select the variables from the first levels of the decision trees built using several split criteria. The aim of this work is to analyze which split criterion obtains the most significant variable subset using only one decision tree, with the aim to use this best criterion in future complex selection methods which uses a set of decision trees. We have used the already known split criteria: Info-Gain (IG), Info-Gain Ratio (IGR), and the Gini Index (GIx). We have also added another split criterion based on imprecise probabilities and uncertainty measures (**IIG**).

The studies carried out in this work reveal that the **IIG** criterion is clearly better than the others considering a trade-off accuracy and number of variables selected in the 27 different databases used in the experimental results. This does not mean that its use in the more complex methods that we referenced ([13,14]) was better, although it is logical to think that we will obtain a more significant set of variables using a lower number of decision trees.

In the future, our first aim is to apply the previously mentioned complex criteria using decision trees with the **IIG** split. For this purpose, we want to use it on gene expression databases where there are a large number of attribute variables and variable selection allows us to extract knowledge as it is possible to identify the most relevant genes that induce a given disease.

References

1. Abellán, J.: Uncertainty measures on probability intervals from Imprecise Dirichlet model. Int. J. of General Systems 35(5), 509–528 (2006)
2. Abellán, J., Moral, S.: Maximum entropy for credal sets. Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems 11, 587–597 (2003)
3. Abellán, J., Moral, S.: Building classification trees using the total uncertainty criterion. Int. J. of Intelligent Systems 18(12), 1215–1225 (2003)
4. Abellán, J., Moral, S.: An algorithm that computes the upper entropy for order-2 capacities. Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems 14(2), 141–154 (2005)
5. Abellán, J., Moral, S.: Upper entropy of credal sets. Applications to credal classification. Int. J. of Approximate Reasoning 39(2-3), 235–255 (2005)
6. Abellán, J., Klir, G.J., Moral, S.: Disaggregated total uncertainty measure for credal sets. Int. J. of General Systems 35(1), 29–44 (2006)
7. Bernard, J.M.: An introduction to the imprecise Dirichlet model for multinomial data. Int. J. of Approximate Reasoning 39, 123–150 (2005)
8. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth Statistics, Probability Series, Belmont (1984)
9. Duda, R.O., Hart, P.E.: Pattern classification and scene analysis. John Wiley and Sons, New York (1973)
10. Fayyad, U.M., Irani, K.B.: Multi-valued interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence, pp. 1022–1027. Morgan Kaufmann, San Francisco (1993)
11. Hall, M.A., Holmes, G.: Benchmarking Attribute Selection Techniques for Discrete Class Data Mining. IEEE TKDE 15(3), 1–16 (2003)
12. Klir, G.J.: Uncertainty and Information: Foundations of Generalized Information Theory. John Wiley, Chichester (2006)
13. Lau, M., Schultz, M.: A Feature Selection Method for Gene Expression Data with Thousands of Features. Technical Report,CS Department, Yale University (2003)
14. Li, J., Liu, H., Ng, S.K., Wong, L.: Discovery of significant rules for classifying cancer diagnosis data. Bioinformatics 19(2), 93–102 (2003)
15. Nadeau, C., Bengio, Y.: Inference for the Generalization Error. Machine Learning (2001)
16. Quinlan, J.R.: Induction of decision trees. Machine Learning 1, 81–106 (1986)
17. Quinlan, J.R.: Programs for Machine Learning. Morgan Kaufmann series in Machine Learning (1993)
18. Ratanamahatana, C., Gunopulos, D.: Feature selection for the naive bayesian classifier using decission trees. App. Art. Intelligence 17, 475–487 (2003)
19. Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal 27, 379–423, 623–656 (1948)
20. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, London (1991)
21. Walley, P.: Inferences from multinomial data: learning about a bag of marbles. J. Roy. Statist. Soc. B 58, 3–57 (1996)
22. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Inference and Learning in Multi-dimensional Bayesian Network Classifiers

Peter R. de Waal and Linda C. van der Gaag

Department of Information and Computing Sciences, Faculty of Sciences,
Utrecht University, P.O. Box 80.089, 3508TB Utrecht, The Netherlands
{waal,linda}@cs.uu.nl

Abstract. We describe the family of multi-dimensional Bayesian network classifiers which include one or more class variables and multiple feature variables. The family does not require that every feature variable is modelled as being dependent on every class variable, which results in better modelling capabilities than families of models with a single class variable. For the family of multi-dimensional classifiers, we address the complexity of the classification problem and show that it can be solved in polynomial time for classifiers with a graphical structure of bounded treewidth over their feature variables and a restricted number of class variables. We further describe the learning problem for the subfamily of fully polytree-augmented multi-dimensional classifiers and show that its computational complexity is polynomial in the number of feature variables.

1 Introduction

Many real-life problems can be viewed as classification problems, in which an instance described by a number of features has to be classified in one of several distinct classes. Bayesian network classifiers have gained considerable popularity for solving such classification problems. The success of especially naive Bayesian classifiers and the more expressive tree-augmented network classifiers is readily explained from their ease of construction from data and their generally good performance. Not all classification problems are one-dimensional, however: in many problems an instance has to be assigned to a most likely combination of classes instead of to a single class. In our application in oncology, for example, we have to classify an oesophageal tumour in terms of its depth of invasion, its spread to lymph nodes, and whether or not it has given rise to haematogenous metastases [5]; in another application, in veterinary medicine, we have to establish a diagnosis for a pig herd in which multiple diseases may be present. Since the number of class variables in a traditional Bayesian network classifier is restricted to one, such classification problems cannot be modelled straightforwardly. One approach is to construct a compound class variable that models all possible combinations of classes. The class variable may then easily end up with an inhibitively large number of values. Moreover, the structure of the problem is not properly reflected in the model. Another approach is to develop multiple classifiers, one for each original class. Multiple classifiers, however, cannot model interaction effects among the various classes. As a consequence they may imply a combination of marginal classifications that does not constitute a most likely explanation of the observed features.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 501–511, 2007.

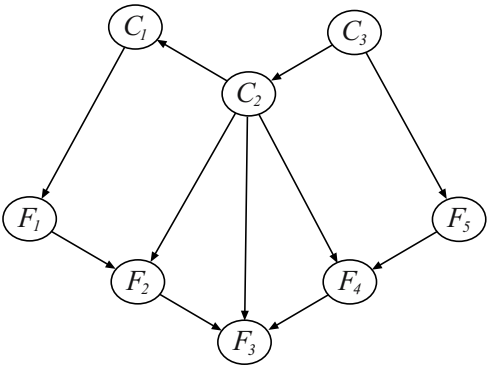


Fig. 1. An example multi-dimensional Bayesian network classifier with class variables C_i and feature variables F_j

In a recent workshop paper, we introduced the family of multi-dimensional Bayesian network classifiers to provide for modelling classification problems in which instances can be assigned to multiple classes [6]. A multi-dimensional Bayesian network classifier includes one or more class variables and multiple feature variables. It models the relationships between the variables by acyclic directed graphs over the class variables and over the feature variables respectively, and connects the two sets of variables by means of a bi-partite directed graph; an example multi-dimensional classifier is depicted in Figure 1. As for one-dimensional Bayesian network classifiers, we distinguished between different types of multi-dimensional classifier by imposing restrictions on their graphical structure. A fully tree-augmented multi-dimensional classifier, for example, has directed trees over its class variables and over its feature variables.

In the present paper, we address the computational complexity of the classification problem for multi-dimensional Bayesian network classifiers. Classification with a multi-dimensional classifier amounts to solving the maximum probability assignment (MPA) problem, which is known to be NP-hard in general [1]. We show, however, that the classification problem can be solved in polynomial time if the graphical structure over the classifier's feature variables has bounded treewidth and the number of class variables is restricted.

Having studied the learning problem for fully tree-augmented multi-dimensional classifiers in our earlier workshop paper, we now address learning fully polytree-augmented classifiers. These classifiers have polytree structures over their class variables and over their feature variables respectively. We present an algorithm for learning these polytree structures from data. Our algorithm is polynomial in the number of feature variables involved and is guaranteed to exactly recover the classifier's graphical structure if the available data reflect the independences of a polytree-augmented multi-dimensional model.

The paper is organised as follows. In Section 2, we briefly review the most commonly used Bayesian network classifiers. In Section 3, we describe the family of

multi-dimensional classifiers and introduce some of its subfamilies. In Section 4 we address the complexity of classification with a multi-dimensional classifier. In Section 5, we study the subfamily of fully polytree-augmented multi-dimensional classifiers and show how the polytree structures over the class variables and over the feature variables can be recovered from data. The paper is rounded off with our concluding observations in Section 6.

2 Preliminaries

Before reviewing the most commonly used Bayesian network classifiers, we introduce our notational conventions. We consider Bayesian networks over a finite set $V = \{X_1, \ldots, X_k\}$, $k \geq 1$, of discrete random variables, where each X_i takes a value in a finite set $Val(X_i)$. For a subset of variables $Y \subseteq V$ we use $Val(Y) = \times_{X_i \in Y} Val(X_i)$ to denote the set of joint value assignments to Y. A Bayesian network now is a pair $B = \langle G, \Theta \rangle$, where G is an acyclic directed graph whose vertices correspond to the random variables and Θ is a set of parameters; the set Θ includes a parameter $\theta_{x_i|\Pi x_i}$ for each value $x_i \in Val(X_i)$ and each joint value assignment $\Pi x_i \in Val(\Pi X_i)$ to the set ΠX_i of parents of X_i in G. The network B defines a joint probability distribution P_B over V which factorizes as $P_B(X_1, \ldots, X_k) = \prod_{i=1}^{k} \theta_{X_i|\Pi X_i}$.

Bayesian network classifiers are Bayesian networks that are tailored to solving problems in which instances described by a number of features have to be classified in one of several distinct classes. The set of random variables V of a Bayesian network classifier is partitioned to this end into a set $V_F = \{F_1, \ldots, F_m\}$, $m \geq 1$, of *feature variables* and a singleton set $V_C = \{C\}$ with the *class variable*. By imposing restrictions on the network's graphical structure, a number of different types of classifier are defined. A naive Bayesian classifier, for example, has a directed tree for its graph G, in which the class variable C is the unique root and each feature variable F_i has C for its only parent. Since its graph has a fixed topology, learning a naive Bayesian classifier amounts to establishing maximum-likelihood estimates from the available data for its parameters θ_C and $\theta_{F_i|C}$, $i = 1, \ldots, m$.

A tree-augmented network (TAN) classifier allows limited conditional dependence among its feature variables. It has the structure of a naive Bayesian classifier, but it allows each feature variable to have at most one other feature variable as a parent; the subgraph induced by the set of feature variables, moreover, is a directed tree. Learning a TAN classifier amounts to determining a tree over the feature variables of maximum likelihood given the available data, and establishing estimates for its parameters. The maximum-likelihood tree of the classifier is readily constructed by a maximum-weight spanning tree algorithm [4]. The notion of allowing a limited form of dependence between the feature variables has been generalised to k-dependence Bayesian (kdB) classifiers [10]. A kdB classifier also has the structure of a naive Bayesian classifier, but it allows each feature variable F_i to have a maximum of k feature variables for its parents. Note that the family of kdB classifiers includes the subfamily of naive Bayesian classifiers and the subfamily of TAN classifiers. An efficient heuristic algorithm for constructing kdB classifiers from data is available [10]; to the best of our knowledge, there is no exact algorithm to determine a maximum-likelihood structure over the

feature variables for $k \geq 2$. Yet another approach to allowing limited dependence among the feature variables has resulted in forest-augmented network (FAN) classifiers [8]. In a FAN classifier, exactly k feature variables are allowed to depend on another feature variable; the subgraph induced by the set of feature variables then is a forest containing exactly k arcs. A maximum-likelihood forest again is constructed by a maximum-weight spanning-tree algorithm [4].

3 Multi-dimensional Classifiers

The various types of Bayesian network classifier reviewed above include a single class variable and as such are one-dimensional. We now describe a family of Bayesian network classifiers that may include multiple class variables [6].

Definition 1. *A* multi-dimensional Bayesian network classifier *is a Bayesian network* $B = \langle G, \Theta \rangle$ *of which the graphical structure equals*

$$G = \langle V_C \cup V_F, A_C \cup A_F \cup A_{CF} \rangle,$$

where

- $V_C = \{C_1, \ldots, C_n\}$, $n \geq 1$, *is the set of* class variables *and* $V_F = \{F_1, \ldots, F_m\}$, $m \geq 1$, *is the set of* feature variables*;*
- $A_C \subseteq V_C \times V_C$ *is the set of arcs between the class variables and* $A_F \subseteq V_F \times V_F$ *is the set of arcs between the feature variables;*
- $A_{CF} \subseteq V_C \times V_F$ *is the set of arcs from the class variables to the feature variables such that for each* $F_i \in V_F$ *there is a* $C_j \in V_C$ *with* $(C_j, F_i) \in A_{CF}$ *and for each* $C_i \in V_C$ *there is an* $F_j \in V_F$ *with* $(C_i, F_j) \in A_{CF}$.

The subgraph $G_C = \langle V_C, A_C \rangle$ *of* G *is called the classifier's* class subgraph*; the subgraph* $G_F = \langle V_F, A_F \rangle$ *is called its* feature subgraph. *The subgraph* $G_{CF} = \langle V, A_{CF} \rangle$ *is called the* feature selection subgraph *of the classifier.*

Within the family of multi-dimensional Bayesian network classifiers, again different types of classifier are distinguished based upon their graphical structures [6]. In a fully naive multi-dimensional classifier, for example, both the class subgraph and the feature subgraph are empty. This subfamily of bi-partite classifiers includes the one-dimensional naive Bayesian classifier as a special case; reversely, any such bi-partite classifier has an equivalent naive Bayesian classifier with a single compound class variable. Another type of multi-dimensional classifier is the fully tree-augmented multi-dimensional classifier in which both the class subgraph and the feature subgraph are directed trees. We further distinguish the subfamily of multi-dimensional classifiers in which the class and feature subgraphs are polytrees, that is, are singly connected. We refer to these classifiers as *fully polytree-augmented multi-dimensional classifiers*.

4 The Complexity of Classification

Classification with a one-dimensional Bayesian network classifier amounts to establishing a value of highest probability for the class variable. Finding such a value is

equivalent to computing the posterior probability distribution over the class variable. This problem is known to be NP-hard in general [3], yet can be solved in polynomial time for Bayesian networks of bounded treewidth. Since naive Bayesian classifiers and TAN classifiers have very small treewidth, for example, classification with these models is performed in polynomial time.

Classification with a multi-dimensional classifier amounts to finding a joint value assignment of highest posterior probability for the set of class variables. Finding such an assignment, given values for all feature variables involved, is equivalent to solving the maximum probability assignment, or MPA, problem. This problem is also known to be NP-hard in general [1], and can also be solved in polynomial time for Bayesian networks of bounded treewidth. Unfortunately, multi-dimensional Bayesian network classifiers can have large treewidths, depending upon the topological properties of their various subgraphs. Building upon a result for undirected bi-partite graphs [2], we have the following upper bound on the treewidth of a multi-dimensional classifier.

Theorem 1. *Let $G = \langle V_C \cup V_F, A_C \cup A_F \cup A_{CF} \rangle$ be the graphical structure of a multi-dimensional classifier and let G_F be its feature subgraph. Then,*

$$\mathit{treewidth}(G) \leq \mathit{treewidth}(G_F) + |\, V_C \,|,$$

where $\mathit{treewidth}(H)$ of a directed graph H is taken to be the treewidth of its moralisation H^m, that is, of the undirected graph obtained from H by connecting the sets of parents of each variable and subsequently dropping all directions.

Proof. We consider the feature subgraph G_F of the classifier, and its moralisation G_F^m. Let $\mathcal{T} = \{Cl_i \mid i = 1, \ldots, n\}$ be a tree decomposition of G_F^m with $n = \mathit{treewidth}(G_F)$, where each element $Cl_i \subseteq V_F$ constitutes a clique in the decomposition. Now let $\mathcal{T}' = \{Cl_i \cup V_C \mid Cl_i \in \mathcal{T}\}$ be the set that is obtained by adding all class variables to each element Cl_i of the decomposition $\mathcal{T}$. Then, $\mathcal{T}'$ is a tree decomposition of the moralisation G^m of G, with treewidth $\mathit{treewidth}(G_F) + |\, V_C \,|$. The property stated in the theorem now follows. □

From the theorem we conclude that the classification problem for a multi-dimensional classifier can be solved in polynomial time if the treewidth of the feature subgraph is bounded and the number of class variables is restricted. We observe that for most applications the number of class variables indeed is much smaller than the number of feature variables. The number of class variables can in fact often be considered constant in terms of the number of feature variables. The connectivity of the class subgraph then is irrelevant for the feasibility of classification. The theorem's proof further shows that the treewidth of a multi-dimensional classifier attains its maximum with a full bi-partite feature selection subgraph. The treewidth then grows linearly with the number of class variables, regardless of the treewidth of the class subgraph. If the feature selection subgraph is not a full directed bi-partite graph, however, the classifier's treewidth may grow at a lesser rate, which depends not just on the topological properties of the feature selection subgraph but of those of the class subgraph as well.

5 Recovery of Fully Polytree-Augmented Classifiers

Within the family of multi-dimensional Bayesian network classifiers, we distinguished between different types of model. In our earlier workshop paper, we focused on the subfamily of fully tree-augmented multi-dimensional classifiers and presented a polynomial algorithm for constructing maximum-likelihood directed trees over the class variables and over the feature variables of such a classifier. In the remainder of the present paper, we focus on the subfamily of fully polytree-augmented multi-dimensional classifiers and study the learning problem for this subfamily. We will show more specifically that previous results of polytree recovery can be extended to construct polytrees over the class variables and over the feature variables of a multi-dimensional classifier.

The recovery of polytree structures has been addressed before by Rebane and Pearl who introduce polytrees as a subfamily of graphical models [9]. They show that, if a probability distribution has a polytree for a perfect map, then this polytree can be recovered optimally from the distribution. By optimal recovery they mean that the underlying undirected graph, or *skeleton*, of the polytree can be recovered precisely and the directions of all edges in the causal basins of the polytree can be determined uniquely, where a causal basin is a sub-polytree composed of a node with more than one parent plus the descendants of this node and all the direct parents of these descendants. Note that the recovered polytree thus in essence is a partially directed skeleton which defines an equivalence class of polytrees modelling the same independence relation.

Before showing how the results of Rebane and Pearl can be extended to provide for the recovery of the polytrees of a fully polytree-augmented classifier, we introduce some notations. For any mutually disjoint sets of variables X, Y, Z with $X, Y \neq \varnothing$, we use $(X \mid Z \mid Y)_G$ to indicate that the sets X and Y are d-separated by the set Z in the directed graph G; $\neg(X \mid Z \mid Y)_G$ then indicates that X and Y are not d-separated by Z in G. We further use $X \perp_P Y \mid Z$ to denote that, under probability distribution P, the sets X and Y are independent given Z; $X \not\perp_P Y \mid Z$ denotes that X and Y are dependent given Z under P.

The following lemma now shows that for the class variables of a multi-dimensional classifier, d-separation restricted to the class subgraph is equivalent to d-separation in the classifier's entire graphical structure.

Lemma 1. *Let $G = \langle V, A\rangle$ be the graphical structure of a multi-dimensional classifier and let $G_C = \langle V_C, A_C\rangle$ be its class subgraph. For any mutually disjoint $X, Y, Z \subseteq V_C$ with $X, Y \neq \varnothing$, we have that $(X \mid Z \mid Y)_{G_C}$ if and only if $(X \mid Z \mid Y)_G$.*

Proof. Let $X, Y, Z \subseteq V_C$ with $X, Y \neq \varnothing$. To prove the property stated in the lemma, we distinguish between the two cases where $Z = \varnothing$ and $Z \neq \varnothing$.

To show that $(X \mid \varnothing \mid Y)_{G_C}$ if and only if $(X \mid \varnothing \mid Y)_G$, we first assume that $(X \mid \varnothing \mid Y)_{G_C}$. We consider a path from X to Y in G. If this path contains class variables only, it is blocked by $\varnothing$ in G_C by the assumption. The path then is also blocked by $\varnothing$ in G. If the path contains a feature variable, it includes at least one converging node. It thus is blocked by $\varnothing$ in G. We conclude that $(X \mid \varnothing \mid Y)_G$. Now assume that $\neg(X \mid \varnothing \mid Y)_{G_C}$. From the assumption, we have that there exists at least one path from X to Y in G_C that is not blocked by $\varnothing$. This path remains unblocked by $\varnothing$ if we extend the graph from G_C to G. We conclude that $\neg(X \mid \varnothing \mid Y)_G$.

To show for $Z \neq \varnothing$ that $(X \mid Z \mid Y)_{G_C}$ if and only if $(X \mid Z \mid Y)_G$, we first assume that $(X \mid Z \mid Y)_{G_C}$. We consider a path from X to Y in G. If this path contains class variables only, it is blocked by Z in G_C by the assumption. It then is also blocked by Z in G. If the path contains a feature variable, then this feature variable is a converging node on the path. Since Z includes class variables only, neither this feature variable nor its descendants are included in Z. The path therefore is also blocked by Z in G. Now assume that $\neg(X \mid Z \mid Y)_{G_C}$. From the assumption, we have that there exists a path from X to Y in G_C that is not blocked by Z. This path is also not blocked by Z in G, from which we conclude that $\neg(X \mid Z \mid Y)_G$. □

The importance of the previous lemma lies in the following corollary which essentially states that if a probability distribution has a multi-dimensional classifier for a perfect map, then its marginal distribution over the class variables has the classifier's class subgraph for a perfect map.

Corollary 1. *Let P be a probability distribution that is representable by a multi-dimensional classifier $G = \langle V, A\rangle$ with the class subgraph $G_C = \langle V_C, A_C\rangle$. Let P_C be the marginal distribution of P over the class variables V_C. Then, P_C is representable by the class subgraph G_C.*

From the corollary we have that the polytree class subgraph of a fully polytree-augmented multi-dimensional classifier can be recovered without having to consider the feature variables. Based upon this observation, we propose the following algorithm for recovering the polytree class subgraph. The algorithm is based on the Generating Polytree recovery algorithm from Rebane and Pearl [9].

Class Polytree Recovery (CPR) Algorithm

1. For every two class variables $C_i, C_j \in V_C$, $C_i \neq C_j$, compute the mutual information $I(C_i; C_j)$ as a weight for the edge between C_i and C_j, where for any two variables X and Y the term $I(X;Y)$ is defined as

$$I(X;Y) = \sum_{x,y} P(x,y) \cdot \log \frac{P(x,y)}{P(x)P(y)},$$

   with $P(X,Y)$ denoting the joint distribution of X and Y.

2. Using Kruskal's algorithm [7] with the weights from Step 1, construct an undirected maximum weighted spanning tree over the class variables V_C.

3. Using the mutual-information terms from Step 1, determine, from the terms equal to zero, the multi-parent variables in the tree of Step 2 and establish the directions for the edges in the resulting causal basins.

The following theorem states that if a probability distribution has a multi-dimensional classifier with a polytree class subgraph for a perfect map, then this polytree is recovered by the CPR algorithm in the optimal sense of Rebane and Pearl.

Theorem 2. *If a probability distribution P is representable by a multi-dimensional classifier with a polytree class subgraph, then the CPR algorithm serves to optimally recover this polytree class subgraph from the distribution.*

Proof. We consider a probability distribution P that is representable by a multi-dimensional classifier with a polytree class subgraph $G_C = \langle V_C, A_C \rangle$. From Corollary 1 we have that G_C is a perfect map for the marginal distribution P_C of P over the class variables V_C. The property stated in the theorem now results from applying Theorems 1 and 2 from [9] to the marginal distribution P_C. □

Note that the above theorem attaches the condition that the probability distribution is representable by a multi-dimensional classifier with a polytree class subgraph, to the recovery of this subgraph by the CPR algorithm. Rebane and Pearl need the same condition to guarantee optimal polytree recovery for their algorithm. If the condition of representability is not met by a probability distribution under study, then no guarantees can be given for the partially directed polytree that is recovered by their algorithm. A similar observation holds for our algorithm, albeit that a slightly weaker condition could be formulated: for the CPR algorithm optimal recovery is also guaranteed if the probability distribution P under study has a multi-dimensional classifier with a polytree class subgraph for an I-map such that this class subgraph is a perfect map for the marginal distribution of P over the class variables.

Having studied the recovery of the polytree class subgraph of a fully polytree-augmented multi-dimensional classifier, we now turn to the recovery of its polytree feature subgraph. We begin by presenting an algorithm to this end.

Feature Polytree Recovery (FPR) Algorithm

1. For every two feature variables $F_i, F_j \in V_F$, $F_i \neq F_j$, compute the conditional mutual information $I(F_i; F_j \mid V_C)$ given the class variables V_C as a weight for the edge between F_i and F_j, where for any two variables X and Y the term $I(X; Y \mid Z)$ given the variables Z, is defined as

$$I(X; Y \mid Z) = \sum_{x,y,z} P(x, y, z) \cdot \log \frac{P(x, y \mid z)}{P(x \mid z) \cdot P(y \mid z)},$$

   with $P(X, Y, Z)$ denoting the joint distribution of X, Y and Z.

2. Using Kruskal's algorithm with the weights from Step 1, construct an undirected maximum weighted spanning tree over the feature variables V_F.

3. Using the conditional mutual information terms from Step 1, determine, from the terms equal to zero, the multi-parent variables in the tree from Step 2 and establish the directions for the edges in the resulting causal basins.

The following theorem now states that the FPR algorithm optimally recovers the polytree feature subgraph of a multi-dimensional classifier with such a subgraph. Note that, in contrast with the class subgraph, the recovery of the feature subgraph cannot be studied without considering the class variables. More specifically, if a probability distribution has a multi-dimensional classifier with a polytree feature subgraph for a perfect

map, then it is not guaranteed that this subgraph is a perfect map for the marginal distribution over the feature variables.

Theorem 3. *If a probability distribution P is representable by a multi-dimensional classifier with a polytree feature subgraph, then the FPR algorithm serves to optimally recover this polytree feature subgraph from the distribution.*

Proof. We first show that Step 2 of the FPR algorithm correctly recovers the skeleton of the polytree feature subgraph of the classifier. We consider three different feature variables $X, Y, Z \in V_F$ for which $X \perp_P Y \mid \{Z\} \cup V_C$ under the distribution P. Analogous to the proof of Theorem 1 from [9], this implies

$$\min \Big[I(X; Z \mid V_C), I(Z; Y \mid V_C) \Big] > I(X; Y \mid V_C).$$

We observe that the inequality also holds if $X \perp_P Y \mid V_C$, since then $I(X; Y \mid V_C) = 0$. The inequality therefore holds for any triplet of variables $X - Z - Y$ which are adjacent in the polytree feature subgraph G_F of the classifier. From this observation, we have that Kruskal's algorithm will never include an edge between any two feature variables that are not directly linked in this polytree. We conclude that Step 2 of the FPR algorithm returns the skeleton of the polytree.

The proof that Step 3 of the algorithm correctly finds the directions of the edges in the causal basins of the polytree feature subgraph, is immediate from the proof of Theorem 2 in [9], once we observe that the adjacent triplet $X \rightarrow Z \leftarrow Y$ is characterised by $X \perp_P Y \mid V_C$, while triplets $X \rightarrow Z \rightarrow Y$ and $X \leftarrow Z \rightarrow Y$ are characterised by $X \not\perp_P Y \mid V_C$. □

The CPR and FPR algorithms presented above constitute the main procedures of learning fully polytree-augmented multi-dimensional classifiers from data. From Theorems 2 and 3, we have that the algorithms are guaranteed to find the classifier's polytree subgraphs if the available data reflect the independences of a polytree-augmented multi-dimensional model.

The overall learning algorithm has a computational complexity that is polynomial in the number of feature variables, for classifiers in which the number of class variables is constant in terms of the number of feature variables. The most costly step in the learning process is Step 2 of the FPR algorithm in which conditional mutual-information terms are computed for all pairs of feature variables. The larger the set $Val(V_C)$ of joint value assignments to the class variables V_C, the more costly the computations involved in the recovery of the feature subgraph become. Moreover, a larger set of joint value assignments $Val(V_C)$ limits the practicability of the FPR algorithm as presented above, especially for applications in which relatively few data are available. In some cases, however, it suffices to condition the mutual-information terms on a strict subset of the set of class variables V_C. We consider as an example the graphical structure from Figure 1 and assume that it is a perfect map for a distribution P. From the graph, it is apparent that $F_i \perp_P F_j \mid \{C_2\}$ for any combination of feature variables with $i = 1, 2$ and $j = 4, 5$. In combination with the inequality from the proof of Theorem 3, this observation implies that the FPR algorithm can recover the polytree feature subgraph using the weights $I(F_i; F_j \mid \{C_2\})$. Establishing whether conditioning on a smaller set

of class variables would suffice for recovering the feature subgraph in general, is an issue for further research.

To conclude, we recall that our earlier algorithm for learning fully tree-augmented classifiers also involves the computation of conditional mutual-information terms [6]. We would like to note here that the computed conditional mutual-information terms play different roles in the two learning algorithms. In the construction of a tree feature subgraph the weights play just a quantitative role, in that the algorithm favours the inclusion of arcs with large weights. In the recovery of a polytree feature subgraph as presented above, the weights also play a qualitative role, in that the algorithm uses the property that zero weights between variables are equivalent to (conditional) independence.

6 Conclusions

We investigated the family of multi-dimensional Bayesian network classifiers which include one or more class variables and multiple feature variables. More specifically, we studied the computational complexity of the classification problem for this family and showed that it can be solved in polynomial time for classifiers with a graphical structure of bounded treewidth over the feature variables and with a limited number of class variables. We further described the learning problem for the subfamily of fully polytree-augmented multi-dimensional classifiers and showed that the polytree structures over a classifier's class variables and feature variables can be recovered from data which correctly reflects the independencies of a polytree-augmented classifier. Our algorithm for this purpose has a complexity that is polynomial in the number of variables involved.

Preliminary experimental results have hinted at the benefits of multi-dimensional classifiers. In experiments on small data sets, the multi-dimensional classifiers provided higher accuracy than their one-dimensional counterparts. In combination with feature selection they also led to classifiers with fewer parameters for the conditional probability tables. In the near future we will conduct a more extensive experimental study in which we test the performance of our multi-dimensional classifiers on larger data sets. We will then study various subfamilies of classifiers with different types of graphical structure. For these families, we will also investigate the treewidths found in practice to assess the practicability of using multi-dimensional classifiers in real-life settings.

References

1. Bodlaender, H.L., van den Eijkhof, F., van der Gaag, L.C.: On the complexity of the MPA problem in probabilistic networks. In: Proceedings of the 15th European Conference on Artificial Intelligence, pp. 675–679. IOS Press, Amsterdam (2002)
2. Bodlaender, H.L., Möhring, R.H.: The pathwidth and treewidth of cographs. Siam Journal of Discrete Methods 6, 181–188 (1993)
3. Cooper, G.F.: The computational complexity of probabilistic inference using Bayesian belief networks. Artificial Intelligence 42, 393–405 (1990)
4. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning 29, 131–163 (1997)
5. van der Gaag, L.C., Renooij, S., Witteman, C.L.M., Aleman, B.M.P., Taal, B.G.: Probabilities for a probabilistic network: a case study in oesophageal cancer. Artificial Intelligence in Medicine 25, 123–148 (2002)

6. van der Gaag, L.C., de Waal, P.R.: Multi-dimensional Bayesian network classifiers. In: Studeny, M., Vomlel, J. (eds.) Proceedings of the Third European Workshop in Probabilistic Graphical Models, Prague, pp. 107–114 (2006)
7. Kruskal, J.B.: On the shortest spanning subtree of a graph and the travelling salesman problem. Proc. Amer. Math. Soc. 7, 48–50 (1956)
8. Lucas, P.J.F.: Restricted Bayesian network structure learning. In: Gámez, G.A., Moral, S., Salmerón, A. (eds.) Advances in Bayesian Network, Studies in Fuzziness and Soft Computing, vol. 146, pp. 217–232. Springer, Heidelberg (2004)
9. Rebane, G., Pearl, J.: The recovery of causal poly-trees from statistical data. In: Kanal, L.N., Levitt, T.S., Lemmer, J.F. (eds.) Proceedings of the 3rd Conference on Uncertainty in Artificial Intelligence, Elsevier, Amsterdam (1989)
10. Sahami, M.: Learning limited dependence Bayesian classifiers. In: KDD-96, Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 335–338. AAAI Press, Menlo Park (1996)

Combining Decision Trees Based on Imprecise Probabilities and Uncertainty Measures*

Joaquín Abellán and Andrés R. Masegosa

Department of Computer Science and
Artificial Intelligence.
University of Granada, Spain
{jabellan,andrew}@decsai.ugr.es

Abstract. In this article, we shall present a method for combining classification trees obtained by a simple method from the imprecise Dirichlet model (IDM) and uncertainty measures on closed and convex sets of probability distributions, otherwise known as credal sets. Our combine method has principally two characteristics: it obtains a high percentage of correct classifications using a few number of classification trees and it can be parallelized to apply on very large databases.

Keywords: Imprecise probabilities, credal sets, uncertainty, entropy, classification, imprecise Dirichlet model.

1 Introduction

Until recently, the fundamental tool used in classification (an important problem in the field of *machine learning*) was the classic probability theory. In this article, our starting point shall be the classification methods which use closed and convex sets of probability distributions, otherwise known as imprecise probabilities or credal sets, and uncertainty measures on these. The problem of classification may generally be defined in the following way: we have a set of observations, called the training set, and we wish to obtain a set of laws in order to assign a value of the variable to be classified (also called class variable) to each new observation. The set used to verify the quality of this set of laws is also called the test set. Classification has important applications in medicine, character recognition, astronomy, banking, etc. A classifier may be represented using a Bayesian network, a neural network, a classification tree, etc. Normally, these methods use the probability theory in order to estimate the parameters with a stopping criterion in order to limit the complexity of the classifier and to annul the dependence of the results with the training database.

* This work has been supported by the Spanish Ministry of Science and Technology under the projects TIN2005-02516 and TIN2004-06204-C03-02; and FPU scholarship programme (AP2004-4678).

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 512–523, 2007.

We shall use the theory of imprecise probabilities (Walley [27]) in order to construct a classification tree. This structure is easy to understand and an efficient classifier. It has its origin in Quinlan's ID3 algorithm [23]. As a basic reference, we should mention the book by Breiman et al. [11]. We shall use the imprecise Dirichlet model [28] (IDM) in order to estimate the probabilities of membership to the respective classes defined by the class variable.

In Abellán and Moral [2,4], we studied how to quantify the uncertainty of a set of probabilities by extending the measures which are used in the theory of evidence of Dempster [12] and Shafer [25] (DST). We shall consider two main origins of uncertainty: conflict and non-specificity. In Abellán and Moral [3,4] and Abellán, Klir and Moral [9], we present functions which verify the fundamental properties that this type of function must verify on more general theories than DST (Dubois and Prade [13], Klir and Wierman [21]).

In the original method of classification of Abellán and Moral [5], to build a classification tree we begin with an empty tree and in order to branch at each node, we select the variable with the greatest degree of total uncertainty reduction in relation to the variable to be classified. In the theory of probability, branching always implies a reduction in entropy. It is therefore necessary to include an additional criterion so as not to create excessively complex models with data dependence. With credal sets, although the conflict produced by branching is smaller, the non-specificity is greater. The stopping criterion is very simple: when branching produces an increase in uncertainty (a decrease in conflict is not offset by an increase in non-specificity). Finally, we shall use a frequency criterion in order to obtain the value of the variable to be classified in the corresponding leaf.

As we say above, to represent the information of a sample for a class variable, we use the IDM, that represents an especial case of credal sets. In Abellán [1], it can be seen that intervals from the IDM can be represented by belief functions. Then, we can say that our original procedure [5] could be considered as a *belief decision tree*. In the literature, we can found others works about belief decision trees, as the one of Elouedi et al. [15] and the one of Vannoorenberghe [26].

In this article, we shall develop a first experimental study of a procedure which combines various trees obtained by simple classification trees. These trees shall be obtained as in the simple method defined in Abellán and Moral [5], with a variation of the system for choosing the variable which we shall use as the root node. For this aim, we shall create a sequence of variables which are more informative than the variable to be classified by default and we shall use each one as the root node in each tree. The rest of the process for completing each tree shall be similar to that described in [5]. Finally, we shall proceed to classify the new cases by considering all the results obtained in each tree. We shall see how the results obtained in this first tree combination work are quite promising.

For our experimentation, we will use a wide set of databases which are commonly used in classification tasks. To check the results obtained, as reference, we compared them with the ones of the known classification method of Naive

Bayes [14] and with the ones of an improved version of C4.5 method of Quinlan [24].

In Section 2 of this article, we shall present necessary notations and definitions. In Section 3, we shall describe the method for combining simple trees in detail. In Section 4, we shall check our procedure with a series of databases which are widely used in classification. Section 5 is devoted to the conclusions and future works.

2 Previous Knowledge

2.1 Naive Bayes

The success of the model developed by Duda and Hart [14] is mainly due to its simplicity, efficiency and effectiveness in classification problems. Before describing the classifier, we will probabilistically describe the supervised classification problem.

Let $\mathcal{D}$ be a database, with size N, and with values in a set $\mathcal{L}$ of (discrete or discretized) attribute variables $\{X_i | \, i = 1, \ldots, r\}$, where each variable has a set of possible states or cases $\Omega_{X_i} = \{x_1^i, x_2^i, ..., x_{|\Omega_{X_i}|}^i\}$, and a class variable C, whose states are $\Omega_C = \{c_1, c_2, ..., c_k\}$. The objective is to obtain information from the database so that given an observation (a set of values of all the attribute variables) it is possible to associate this with a value of the class variable.

If we represent the new sample as $\mathbf{x}$, with $\mathbf{x} = \{x_{h_i}^1, .., x_{h_r}^r\}$. The Naive Bayes predicts value c_i in the following way:

$$arg \max_{c_i} \left(P(c_i | \mathbf{x}) \right).$$

Now, based on the supposition that the attribute variables are independent given the class variable, it can be expressed as

$$arg \max_{c_i} \left(P(c_i) \prod_{j=1}^{r} P(x_{h_j}^j | c_i) \right).$$

2.2 Decision Trees and Split Criteria

A decision tree (or classification tree) is a simple structure that can be used as a classifier. Within a decision tree, each node represents an attribute variable and each branch represents one of the states of this variable. A tree leaf specifies the expected value of the class variable depending on the information contained in the training data set. When we obtain a new sample or instance of the test data set, we can make a decision or prediction about the state of the class variable following the path to the tree from the root until a leaf using the sample values and the tree structure. Associated to each node is the most informative variable which has not already been selected in the path from the root to this node (as long as this variable provides more information than if it had not been included).

In this last case, a leaf node is added with the most probable class value for the partition of the data set defined with the configuration given by the path until the tree root.

In order to measure the quantity of information, several criteria or metrics can be used, and these are called split criteria. In this article, we will focus on the following ones: Info-Gain Ratio (Quinlan [24]) and Imprecise Info-Gain (Abellán and Moral [5]).

Info-Gain Ratio. In order to improve the ID3 model [23], Quinlan introduces the C4.5 model [24], where the Info-Gain split criterion is replaced by an Info-Gain Ratio criterion which penalizes variables with many states. C4.5 model is defined to work with continuous variables, it is possible to work with missing data, and it has a posterior pruning process that is introduced to improve the results and to obtain less complex structures.

The Info-Gain Ratio of an attribute variable X_i on a class variable C can be expressed as:

$$IGR(X_i, C) = \frac{IG(X_i, C)}{H(X_i)},$$

with

$$IG(X_i, C) = H(C) - H(C|X_i),$$

where $H(C)$ is the entropy of C: $H(C) = -\sum_j p(c_j) \log p(c_j)$, with $p(c_j) = p(C = c_j)$, the probability of each value of the class variable estimated in the training data set.

An improved version of C4.5, called J48, can be obtained in *Weka* software [29].

Imprecise Info-Gain. The Imprecise Info-Gain criterion was first used for building decision trees in Abellán and Moral's method [5]. In a similar way to ID3, this tree is only defined for discrete variables, it cannot work with missing values, and it does not carry out a posterior pruning process. It is based on the application of uncertainty measures on convex sets of probability distributions. More specifically, probability intervals are extracted from the database for each case of the class variable using Walley's imprecise Dirichlet model (IDM) [28], which represents a specific kind of convex sets of probability distributions, and on these the entropy maximum is estimated. This is a total measure which is well known for this type of set (see Abellán, Klir and Moral [9]).

The IDM depends on a hyperparameter s and it estimates that (in a given database) the probabilities for each value of the class variable are within the interval:

$$p(c_j) \in \left[\frac{n_{c_j}}{N + s}, \frac{n_{c_j} + s}{N + s} \right],$$

with n_{c_j} as the frequency of the set of values $(C = c_j)$ in the database. The value of parameter s determines the speed with which the upper and lower probability

values converge when the sample size increases. Higher values of s give a more cautious inference. Walley [28] does not give a definitive recommendation for the value of this parameter but he suggests values between $s = 1$ and $s = 2$. In Bernard [10], we can find reasons in favor of values greater than 1 for s.

If we label $K(C)$ and $K(C|(X_i = x_t^i))$ for the following sets of probability distributions q on Ω_C:

$$K(C) = \left\{ q | \, q(c_j) \in \left[\frac{n_{c_j}}{N+s}, \frac{n_{c_j}+s}{N+s} \right] \right\},$$

$$K(C|(X_i = x_t^i)) = \left\{ q | \, q(c_j) \in \left[\frac{n_{\{c_j, x_t^i\}}}{N+s}, \frac{n_{\{c_j, x_t^i\}}+s}{N+s} \right] \right\},$$

with $n_{\{c_j, x_t^i\}}$ as the frequency of the set of values $\{C = c_j, X_i = x_t^i\}$ in the database, we can define the Imprecise Info-Gain for each variable X_i as:

$$IIG(X_i, C) = S(K(C)) - \sum_t p(x_t^i) S(K(C|(X_i = x_t^i))),$$

where $S()$ is the maximum entropy function of a credal set.

For the previously defined intervals and for a value of s between 1 and 2, it is very easy to obtain the maximum entropy for these using Abellán's procedure [1].

3 Combination of Trees

Database *Monks1* is an artificial dataset which comprises six attribute variables: Headshape, Bodyshape, Issmiling, Holding, Jackecolor and Hastie; and one variable to be classified that reaches two possible states: c_0 when Jackecolor reaches it first possible state or when Headshape=Bodyshape. If we apply our classification tree method using IIG split criterion with k-10 folds cross validation on *Monks1*, we obtain a percentage of correct classifications of 79.5%, but changing the root node with the second more informative variable (by IIG) we can obtain a 100% percentage of correct classifications. This leads us to develop the idea of varying the way the root node is chosen.

We propose a new classification method by combining trees obtained by the simple method described in the previous section. For this, we need to fix a number of trees that we shall provisionally call η which must be lower than the number of attribute variables in the database which are more informative than the variable to be classified. Using the notation from the previous section, we can describe the method in the following way:

1. Obtain the following $\alpha_i j$ values and the attribute variables Z_j:

$$\alpha_1 = \min_{X_i \in \mathcal{L}_1} \{IIG(X_i, C)\}, \quad Z_1 = arg(\alpha_1);$$

$$\alpha_2 = \min_{X_i \in \mathcal{L}_2} \{IIG(X_i, C)\}, \quad Z_2 = arg(\alpha_2);$$

$$\ldots\ldots$$

$$\alpha_\eta = \min_{X_i \in \mathcal{L}_\eta} \{IIG(X_i, C)\}, \quad Z_\eta = arg(\alpha_\eta);$$

where the previous $\mathcal{L}_j$ sets are as follows:

$$\mathcal{L}_1 = \mathcal{L},$$

$$\mathcal{L}_2 = \mathcal{L}_1 - \{Z_1\},$$

$$\mathcal{L}_3 = \mathcal{L}_2 - \{Z_2\},$$

$$\cdots$$

$$\mathcal{L}_\eta = \mathcal{L}_{\eta-1} - \{Z_{\eta-1}\}.$$

2. Using each variable of the set $\{Z_j | \, j = 1, 2, \cdots, \eta\}$ as the root variable, obtain η simple trees using the IIG split criterion.
3. In order to classify a new case, apply the new case to each of the η trees. Let $n^j_{c_i}$ be the frequency obtained for the value of the variable to be classified c_i in the tree j. Obtain the following relative frequencies (probabilities):

$$q_i = \sum_j \frac{n^j_{c_i}}{\sum_{c_i} n^j_{c_i}},$$

for each c_i.
For the new case, the value obtained c_h shall be: $c_h = arg(\max_i\{q_i\})$.

It can be seen how the idea is to create a sequence of variables which improve (whenever possible) the uncertainty by default of the variable to be classified. This sequence helps us to create a set of simple classification trees using these variables as root nodes. In each tree, having chosen the variable that we use as a tree's root node, the procedure continues as in the one originally described. In other words, the procedure only modifies the choice of the root node. When we have the set of trees used for classification, for a new instance, we select the value c_h of the class variable C with greatest sum of probabilities over all the trees.

There are many references in the literature about the strategies for combining classifiers (see [18]). There are several ways to use more than one classifier in a classification problem. An approach consists of combining them through majority voting or different linear and non linear combinations. An important characteristic of the method presented above, as shown in Step 3, is that we use a decision criterion of maximum probability, where the probabilities obtained from each classification tree are considered with the same importance, i.e. they are considered with the same weight.

4 Experimentation

In order to check the above procedure, we have used a wide and different set of 27 known databases, obtained from the *UCI repository of machine learning*

Table 1. Database Description

Database	N	Attrib	Num	Nom	k	Range
Anneal	898	38	6	32	6	2-10
. . . . .						
Audiology	226	69	0	69	24	2-6
Autos$_*$	205	25	15	10	7	2-22
Breast-cancer	286	9	0	9	2	2-13
Colic$_*$	368	22	7	15	2	2-6
Credit-german$_*$	1000	20	7	13	2	2-11
Diabetes-pima$_*$	768	8	8	0	2	-
Glass-2$_*$	163	9	9	0	2	-
Hepatitis$_*$	155	19	4	15	2	2
Hypothyroid$_*$	3772	29	7	22	4	2-4
Ionosfere$_*$	351	35	35	0	2	-
Kr-vs-kp	3196	36	0	36	2	2-3
Labor$_*$	57	16	8	8	2	2-3
Letter$_*$	20000	16	16	0	2	-
. . . . .						

Database	N	Attrib	Num	Nom	k	Range
. . . . .						
Lymph	146	18	3	15	4	2-8
Mfeat-pixel	2000	240	240	0	10	5-7
Mushroom	8123	22	0	22	2	2-12
Optdigits	5620	64	64	0	10	-
Segment$_*$	2310	19	16	0	7	-
Sick$_*$	3772	29	7	22	2	2
Solar-flare1	323	12	0	12	2	2-6
Sonar$_*$	208	60	60	0	2	-
Soybean	683	35	0	35	19	2-7
Sponge	76	44	0	44	3	2-9
Vote	435	16	0	16	2	2
Vowel	990	11	10	1	11	2
Zoo	101	16	1	16	7	2

databases which can be directly downloaded from ftp://ftp.ics.uci.edu/machine-learning-databases. A brief description of these can be found in Table 1, where column "N" is the number of instances in the databases, column "Attrib" is the number of attribute variables, column "Nom" is the number of nominal variables, column "k" is the number of cases or states of the class variable (always a nominal variable) and column "Range" is the range of states of the nominal variables of each database.

For our experimentation, we have used *Weka* software [29] on Java 1.5, and we have added the necessary methods to build decision trees using the IIG split criterion, which has been taken from the *Elvira* platform [16] and to build combined classification trees based on IIG. In this experimentation, we have used the parameter of the IDM $s = 1.5$ because it obtains a better trade-off between size of the tree and accuracy in classification than with the others values between $s = 1$ and $s = 2$.

We have applied the following preprocessing: databases with missing values have been replaced with mean values (for continuous variables) and mode (for discrete variables) using *Weka's* own filters. In the same way, continuous variables have been discretized in five equal frequency intervals. We have not used Fayyad and Irani's known discretization method [17] because for 13 of the 19 databases with continuous variables (these are marked in Table 1 with $*$) this leaves a large number of variables with only one state so it is the same as removing them. Using equal frequency discretization is therefore of no benefit to any of the classification methods presented. We note that the preprocessing has been applied using the training set and then translate it to the test set. For each database, we have repeated 10 times a k-10 folds cross validation procedure.

We would also like to compare the results with other known method, using the same databases and the same preprocessing. In Tables 2 and 3, therefore, we present the results obtained on the databases uses. To use as reference, in Table 2, we have inserted column NB, that corresponds to Naive Bayes percentage of correct classifications and column J48, in Table 3, that corresponds to an improved version of the C4.5 method of Quinlan [24], based on the ID3 [23],

Table 2. Percentage of correct classifications compared with NB

Database	NB	$M1$	$M2$	$M3$	$M4$	$M5$	$M6$
Anneal	93.30	99.51 ○	99.53 ○	99.70 ○	99.50 ○	**99.71** ○	99.70 ○
Audiology	71.40	76.29	76.29	**78.76** ○	78.41 ○	77.71 ○	77.89 ○
Autos	62.32	74.31 ○	77.75 ○	79.64 ○	79.73 ○	80.47 ○	**80.90** ○
Breast-cancer	72.94	73.79	74.29	73.75	74.07	**74.56**	74.42
Colic	77.99	84.75 ○	83.77 ○	83.90 ○	84.12 ○	84.77 ○	**84.80** ○
Credit-german	**75.14**	70.71 ●	70.42 ●	72.02 ●	72.71	73.30	73.55
Diabetes-pima	74.49	**75.42**	74.48	75.13	74.48	74.84	74.41
Glass-2	76.40	73.79	78.98	81.03	**82.31**	82.12	82.07
Hepatitis	**84.91**	81.53	82.28	81.76	81.95	81.45	81.38
Hypothyroid	94.98	96.52 ○	96.32 ○	96.22 ○	96.43 ○	96.48 ○	**96.51** ○
Ionosphere	89.00	89.12	88.86	90.08	90.79	90.91	**90.94**
Kr-vs-kp	87.79	99.20 ○	99.21 ○	99.21 ○	99.26 ○	99.27 ○	**99.33** ○
Labor	**94.73**	87.83	87.83	90.83	91.90	92.90	93.17
Letter	68.00	79.89 ○	82.21 ○	83.30 ○	84.81 ○	85.51 ○	**85.61** ○
Lymph	**86.04**	75.04 ●	75.56 ●	77.95 ●	78.01 ●	78.62 ●	79.37
Mfeat-pixel	**93.36**	79.97 ●	83.25 ●	85.95 ●	87.04 ●	87.90 ●	88.59 ●
Mushroom	95.52	**100.00** ○	**100.00** ○	**100.00** ○	**100.00** ○	**100.00** ○	**100.00** ○
Optdigits	**91.60**	77.47 ●	81.61 ●	86.94 ●	89.49 ●	90.85	90.89
Segment	84.32	92.85 ○	93.35 ○	93.55 ○	**93.61** ○	93.55 ○	**93.61** ○
Sick	92.07	93.63 ○	93.61 ○	93.55 ○	93.59 ○	93.58 ○	**93.77** ○
Solar-flare1	93.02	97.78 ○	97.56 ○	97.72 ○	**97.81** ○	97.78 ○	97.75 ○
Sonar	**77.49**	67.59 ●	71.13	71.52	72.04	72.71	73.26
Soybean	92.20	91.11	91.42	92.24	92.20	91.83	**92.27**
Sponge	92.11	93.89	93.64	**95.00**	**95.00**	**95.00**	94.71
Vote	90.23	95.21 ○	95.91 ○	95.95 ○	**96.16** ○	95.97 ○	96.06 ○
Vowel	60.44	69.30 ○	70.44 ○	74.05 ○	77.47 ○	80.64 ○	**81.53** ○
Zoo	94.08	97.82	**98.11**	98.02	96.93	97.82	97.24
Average	83.92	84.98	85.84	86.96	87.40	87.79	**87.92**
W/T/D		12/10/5	12/11/4	13/10/4	13/11/3	13/12/2	13/13/1

○, ● statistically significant improvement or degradation

which uses a classification tree with classic probabilities[1]. Columns Mi correspond to the results of the classification method which combines i simple trees as described above, being $M1$ the simple classification tree using IIG criterion.

Also, in these Tables, we present a test for statistically significant differences for the accuracy means between the Naive Bayes and for each Mi in Table 2 and between $J48$ and each Mi in Table 3. This test is a corrected two-tailed paired t-test [22] with a 5% significance level. ○ represents a statistically significant improvement in the accuracy mean in relation to the Naive Bayes in Table 2 and also ● a statistically significant deterioration in the accuracy mean in relation to the Naive Bayes (respectively with $J48$ in Table 3). In the last row of this table (W/T/D), we can see the accumulated number of wins (○), ties and defeats (●)

[1] J48 method can be obtained via *Weka* software, available in http://www.cs.waikato.ac.nz/ ml/weka/

Table 3. Percentage of correct classifications compared with J48

Database	$J48$	$M1$	$M2$	$M3$	$M4$	$M5$	$M6$
Anneal	98.68	99.51 ○	99.53 ○	99.70 ○	99.50 ○	**99.71** ○	99.70 ○
Audiology	77.22	76.29	76.29	**78.76**	78.41	77.71	77.89
Autos	78.02	74.31	77.75	79.64	79.73	80.47	**80.90**
Breast-cancer	**75.26**	73.79	74.29	73.75	74.07	74.56	74.42
Colic	**85.78**	84.75	83.77	83.90	84.12	84.77	84.80
Credit-german	71.57	70.71	70.42	72.02	72.71	73.30	**73.55**
Diabetes-pima	75.17	**75.42**	74.48	75.13	74.48	74.84	74.41
Glass-2	76.97	73.79	78.98	81.03	**82.31**	82.12	82.07
Hepatitis	81.18	81.53	**82.28**	81.76	81.95	81.45	81.38
Hypothyroid	**96.86**	96.52	96.32 ●	96.22 ●	96.43 ●	96.48 ●	96.51
Ionosphere	89.58	89.12	88.86	90.08	90.79	90.91	**90.94**
Kr-vs-kp	**99.44**	99.20	99.21	99.21	99.26	99.27	99.33
Labor	88.63	87.83	87.83	90.83	91.90	92.90	**93.17**
Letter	80.89	79.89 ●	82.21 ○	83.30 ○	84.81 ○	85.51 ○	**85.61** ○
Lymph	78.08	75.04	75.56	77.95	78.01	78.62	**79.37**
Mfeat-pixel	78.66	79.97	83.25 ○	85.95 ○	87.04 ○	87.90 ○	**88.59** ○
Mushroom	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
Optdigits	78.98	77.47	81.61 ○	86.94 ○	89.49 ○	90.85 ○	**90.89** ○
Segment	92.64	92.85	93.35	93.55 ○	**93.61** ○	93.55 ○	**93.61** ○
Sick	93.62	93.63	93.61	93.55	93.59	93.58	**93.77**
Solar-flare1	**97.84**	97.78	97.56	97.72	97.81	97.78	97.75
Sonar	71.80	67.59	71.13	71.52	72.04	72.71	**73.26**
Soybean	**92.63**	91.11	91.42	92.24	92.20	91.83	92.27
Sponge	92.50	93.89	93.64	**95.00**	**95.00**	**95.00**	94.71
Vote	**96.27**	95.21	95.91	95.95	96.16	95.97	96.06
Vowel	76.75	69.30 ●	70.44 ●	74.05	77.47	80.64 ○	**81.53** ○
Zoo	92.61	97.82	**98.11**	98.02	96.93	97.82	97.24
Average	85.84	84.98	85.84	86.96	87.40	87.79	**87.92**
W/T/D		1/24/2	4/21/2	5/21/1	5/21/1	6/20/1	6/21/0

○, ● statistically significant improvement or degradation

in the statistical tests for each method and in relation to the Naive Bayes and $J48$, in Tables 2 and 3 respectively.

In Table 4, we can see a comparison of the accumulated number of wins and defeats (and also the difference between these) across the whole series of databases and taking in each case each one of the combined methods Mi, with $i > 1$, and the Naive Bayes, $J48$ and the simple method $M1$. The number of wins and defeats are taken from the previously described statistical tests for a statistically significant difference between the accuracy mean of two classifiers, and the significance level used is 5%.

The choice of the values of $\eta = 2, 3, 4, 5, 6$ is due in part to the set of databases selected for the experiments. As we can see, there are databases with only 8 attribute variables. Evidently, the number of simple trees selected should depend on the number of available attribute variables that are more informative than the class variable. We have taken databases which do have aspects which may

Table 4. Number of Wins (Wi) and Defeats (Di) comparison over the right classification percentage between Mi, with $i > 1$ and NB, $J48$ and $M1$

	W2	D2	W2-D2	W3	D3	W3-D3	W4	D4	W4-D4	W5	D5	W5-D5	W6	D6	W6-D6
NB	12	4	8	13	4	9	13	3	10	13	2	9	13	1	12
$J48$	4	2	2	5	1	4	5	1	4	6	1	5	6	0	6
$M1$	3	0	3	7	0	7	8	0	8	7	0	7	8	0	8

be clearly differentiated between, such as the number of cases of the variable to be classified, database size, number of cases of the attribute variables, etc.

On the basis of the results obtained, we can make the following comments:

- In all the databases the results of the combined method $M2$ are better or similar in average than the ones of the $M1$, NB, $J48$ methods. With only 2 simple classification trees we can equalize the results of a more complex procedure such that $J48$ method and improve the ones of NB and our simple procedure $M1$, as we can see in Tables 2 and 3. With 3, 4, 5 or 6 simple classification trees we can obtain differences in favor of our combined methods Mi, with $i > 1$, as we can see in Table 4.
- In base on Tables 2 and 3, we can say that the choice of 2, 3, 4, 5 or 6 trees to combine could be insufficient when we work with databases with very large sets of attribute variables, therefore we believe that the number of trees should depend on the database to be considered. It would be necessary to continue experimenting to see when increasing the number of trees (after taking the results) decreases the number of correct classifications for each database.

5 Conclusions and Future Works

We have presented a first procedure for the combination of classification trees obtained using the IDM and uncertainty measures. By combining a low number of simple classification trees, this method is able to obtain considerable better percentage of correct classifications than the known Naive Bayes and than an improved version of C4.5 (J48).

Another characteristic of our method is that it can be executed in parallel. Recently, with the availability of large databases in application areas (as in bioinformatics, medical informatics, scientific data analysis, financial analysis, telecommunications, retailing, and marketing) it is becoming increasingly important to execute data mining tasks in parallel. By the definition of our combined procedure, it is possible to obtain each classification tree (changing the root node) in parallel before combining them.

This is a first result and as we have said in the experimentation section, it needs to be studied further. We think that it would be necessary to carry out more experiments with more varied databases and artificial databases in order to: (1) Analyze in proof how the results vary according to the number of trees taken; (2) Ascertain the ideal relationship between the number of treesand the number

of attribute variables or the number of attribute variables more informative than the variable to be classified.

We have created a sequence of attribute variables which improve the uncertainty by default of the variable to be classified. Then, another interesting aspect that the method involves is the possibility of introduce a combination method which take into account the informative value obtained for each root node with respect to the class variable. This value could be considered as a weight of each tree in the combination method.

References

1. Abellán, J.: Uncertainty measures on probability intervals from Imprecise Dirichlet model. Int. J. General Systems 35(5), 509–528 (2006)
2. Abellán, J., Moral, S.: Completing a Total Uncertainty Measure in Dempster-Shafer Theory. Int. J. General Systems 28(4-5), 299–314 (1999)
3. Abellán, J., Moral, S.: A Non-specificity Measure for Convex Sets of Probability Distributions. Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems 8(3), 357–367 (2000)
4. Abellán, J., Moral, S.: Maximum entropy for credal sets. Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems 11(5), 587–597 (2003)
5. Abellán, J., Moral, S.: Building classification trees using the total uncertainty criterion. Int. J. of Intelligent Systems 18(12), 1215–1225 (2003)
6. Abellán, J., Moral, S.: Difference of entropies as a non-specificty measure for credal sets. Int. J. of General Systems 34(3), 201–214 (2005)
7. Abellán, J., Moral, S.: An algorithm that computes the upper entropy for order-2 capacities. Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems 14(2), 141–154 (2006)
8. Abellán, J., Moral, S.: Upper entropy of credal sets. Applications to credal classification. Int. J. of Approximate Reasoning 39(2-3), 235–255 (2005)
9. Abellán, J., Klir, G.J., Moral, S.: Disaggregated total uncertainty measure for credal sets. Int. J. of General Systems 35(1), 29–44 (2006)
10. Bernard, J.M.: An introduction to the imprecise Dirichlet model for multinomial data. Int. J. of Approximate Reasoning 39, 123–150 (2005)
11. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth Statistics, Probability Series, Belmont (1984)
12. Dempster, A.P.: Upper and lower probabilities induced by a multivalued mapping. Ann. Math. Stat. 38, 325–339 (1967)
13. Dubois, D., Prade, H.: Properties and Measures of Information in Evidence and Possibility Theories. Fuzzy Sets and Systems 24, 183–196 (1987)
14. Duda, R.O., Hart, P.E.: Pattern classification and scene analysis. John Wiley and Sons, New York (1973)
15. Elouedi, Z., Mellouli, K., Smets, Ph.: Belief decision trees: theoretical foundations. International Journal of Approximate Reasoning 28(2-3), 91–124 (2001)
16. Elvira: An Environment for Creating and Using Probabilistic Graphical Models. In: proceedings of the First European Workshop on Probabilistic Graphical Models (PGM'02), Cuenca (Spain), pp. 1–11 (2002)
17. Fayyad, U.M., Irani, K.B.: Multi-valued interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence, pp. 1022–1027. Morgan Kaufmann, San Mateo (1993)

18. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(3), 226–239 (1998)
19. Klir, G.J.: Uncertainty and Information: Foundations of Generalized Information Theory. John Wiley, Hoboken (2006)
20. Klir, G.J., Smith, R.M.: On measuring uncertainty and uncertainty-based information: Recent developments. Annals of Mathematics and Artificial Intelligence 32(1-4), 5–33 (2001)
21. Klir, G.J., Wierman, M.J.: Uncertainty-Based Information. Phisica-Verlag, Heidelberg (1998)
22. Nadeau, C., Bengio, Y.: Inference for the Generalization Error. Machine Learning (2001)
23. Quinlan, J.R.: Induction of decision trees. Machine Learning 1, 81–106 (1986)
24. Quinlan, J.R.: Programs for Machine Learning. Morgan Kaufmann series in Machine Learning (1993)
25. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
26. Vannoorenberghe, P.: On aggregating belief decision trees. Information Fusion 5(3), 179–188 (2004)
27. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, London (1991)
28. Walley, P.: Inferences from multinomial data: learning about a bag of marbles. J. Roy. Statist. Soc. B 58, 3–57 (1996)
29. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Belief Classification Approach Based on Generalized Credal EM

Imene Jraidi and Zied Elouedi

LARODEC,
Institut Supérieur de Gestion Tunis,
41 Avenue de la liberté,
2000 Le Bardo, Tunisie
imene_jrd@yahoo.fr,zied.elouedi@gmx.fr

Abstract. The EM algorithm is widely used in supervised and unsupervised classification when applied for mixture model parameter estimation. It has been shown that this method can be applied for partially supervised classification where the knowledge about the class labels of the observations can be imprecise and/or uncertain. In this paper, we propose to generalize this approach to cope with imperfect knowledge at two levels: the attribute values of the observations and their class labels. This knowledge is represented by belief functions as understood in the Transferable Belief Model. We show that this approach can be applied when the data are categorical and generated from multinomial mixtures.

Keywords: Expectation Maximization, mixture models, Transferable Belief Model, partially supervised classification.

1 Introduction

Operating within an imperfect environment and facing imprecise, uncertain and even missing information is the real challenge in decision making. For instance, a doctor has to make a decision even if he is not able to identify the exact disease of his patient but he only knows that the patient has not such kind of diseases. On the other hand, a controller system must be able to integrate multiple sensors even when only a fraction may operate at a given time. In this context, most standard classification methods encounter a real problem to meet these real life situations which make them inappropriate to classify objects characterized by such imperfect information.

The idea is thus to combine classification methods with theories managing uncertainty as the belief function theory [11]. In the Transferable Belief Model's interpretation (TBM) [14], this theory provides a formalism for handling subjective and personal judgments and that can also deal with objective probabilities. So, this theory is able to handle partial knowledge and cope with partial and even total ignorance. Besides, this theory has provided a powerful tool to deal with uncertainty in classification problems. We notably mention belief decision trees [5], belief k-nearest neighbor [4], belief K-modes [2], etc.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 524–535, 2007.

On the other side, the Expectation-Maximization (EM) algorithm [3] is a generic approach for parameter estimation in incomplete data problems and has been widely used in supervised and unsupervised classification [7, 8]. In this context, data are assumed to be generated from a mixture model where each component of the mixture or class is identified by a probability distribution. In the supervised mode or discrimination, the class labels of the observations are known a priori and are used to classify new observations with unknown class labels. In the unsupervised mode or clustering, the class labels of the observations are unknown a priori and the goal is to find a partitioning of the observations by grouping similar observations together. Besides, when the class labels are only partially known that is the actual class of the observations can be imprecise or uncertain, the classification procedure becomes partially supervised.

Several works have been proposed in this uncertain context [1, 15]. In [1], the class labels can be imprecise and a probabilistic model relating the imprecise label to the true class is assumed. In [15], the class labels can be imprecise and/or uncertain and this knowledge is represented by belief functions. In both approaches, uncertainty occurs only at the class labels of the observations. However, uncertainty may also appear in the values of the attributes.

We propose then to treat a more general case where uncertainty can arise not only in the class labels but also in the values of the attributes characterizing the observations. This method is based on both the EM approach and the belief function theory as understood in the TBM.

The remainder of the paper is organized as follows. We start by presenting the EM algorithm for learning mixture models. Next, we outline the necessary background concerning the belief function theory and we describe the EM algorithm within this framework. Then, we develop our generalized approach that takes into account uncertainty in the attributes of the observations when data are categorical and generated from multinomial mixtures.

2 The EM Algorithm for Learning Mixture Models

In the mixture modeling approach [9], the data $X = \{x_1, ..., x_n\}$ are assumed to be identically and independently distributed (iid) according to a probability function given by:

$$f(x_i|\Theta) = \sum_{k=1}^{K} \pi_k f_k(x_i|\theta_k) \ , \tag{1}$$

where K is the number of components in the mixture, π_k are the mixing proportions that must be non negative and sum to one, f_k denotes a component, i.e. a probability function parameterized by θ_k, and $\Theta = \{(\pi_k, \theta_k), k = 1, ..., K\}$ are the parameters of the model to be estimated.

In this paper, we treat categorical data generated from multinomial mixture models where each observation x_i is described by D categorical attributes, with a respective number of categories $nb_1, ..., nb_D$. The data X can be represented by n binary vectors $(x_i^{dj}; d = 1, ..., D, j = 1, ..., nb_d)$ where $x_i^{dj} = 1$ if the attribute x_i^d

has the category j and 0 otherwise. In this model, each component k is identified by a D-dimensional multinomial distribution given by:

$$f_k(x_i|\theta_k) = \prod_{d=1}^{D} \prod_{j=1}^{nb_d} (p_k^{dj})^{x_i^{dj}} \ , \tag{2}$$

where the parameters θ_k are given by the probabilities $p_k^{dj}(d = 1, ..., D, j = 1, ..., nb_d)$, that the attribute x_i^d has the category j. In this model, the D variables are assumed to be independent given the component k [6].

2.1 Maximum Likelihood Estimation

To estimate the parameters Θ, we generally apply the Maximum Likelihood Estimation (MLE) principle: the parameters that have most likely generated the data, are those that maximize the likelihood (or the log-likelihood for the sake of simplicity) given by:

$$\mathcal{L}(\Theta|X) = \sum_{i=1}^{n} log(\sum_{k=1}^{K} \pi_k f_k(x_i|\theta_k)) \ . \tag{3}$$

Generally, the maximization of this equation cannot be obtained analytically. The classical approach to solve this problem is the EM algorithm [3] which provides an iterative procedure for computing MLE. In order to use the EM algorithm, the problem has to be reformulated as an incomplete data problem.

2.2 An Incomplete Data Problem

The idea is to introduce a set of "hidden" variables $Z = \{z_1, ..., z_n\}$ that indicate which component of the mixture has generated each observation. The problem would decouple then into a set of simple maximizations. More precisely, $z_i = (z_{i1}, ..., z_{ik}, ...z_{iK})$ where $z_{ik} = 1$ if x_i has been generated from the component k and 0 otherwise. The whole data $Y = \{y_1, ..., y_n\}$ where $(y_i = (x_i, z_i))$, is then the so-called augmented data or complete data. Using these indicator variables Z, the equation (3) can be reformulated as the complete log-likelihood:

$$\mathcal{L}_c(\Theta|Y) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} log(\pi_k f_k(x_i|\theta_k)) \ . \tag{4}$$

2.3 The EM Algorithm

The EM algorithm can now be applied by considering the variables Z as the missing data. The algorithm provides a sequence of estimates $\Theta^{(t)}$[1], of the parameters Θ by the iteration of two steps: an E-step (for Expectation) and a M-step (for Maximization).

[1] We use the subscript (t) to denote the iteration t of the EM algorithm.

E-Step. The E-step computes the conditional expectation of the complete log-likelihood $\mathcal{L}_c(\Theta|Y)$ given the observed data X and the current parameters $\Theta^{(t)}$:

$$Q(\Theta|\Theta^{(t)}) = E[\mathcal{L}_c(\Theta|X,Z)|X,\Theta^{(t)}] \,, \tag{5}$$

which is a linear function of the missing data z_{ik}. So, at the iteration t, the E-step just requires to compute the conditional expectation of z_{ik} given X and $\Theta^{(t)}$:

$$E[z_{ik}|X,\Theta^{(t)}] = t_{ik}^{(t)} \,. \tag{6}$$

Actually, this quantity is nothing else then the posterior probability $p(z_{ik} = 1|X,\Theta^{(t)})$ that the observation x_i has been generated by the component f_k estimated at the iteration t. This probability measure is computed through the Bayes rule as follows:

$$t_{ik}^{(t)} = \frac{\pi_k^{(t)} f_k(x_i|\theta_k^{(t)})}{\sum_{l=1}^{K} \pi_l^{(t)} f_l(x_i|\theta_l^{(t)})} \,. \tag{7}$$

Using this result the equation (5) becomes:

$$Q(\Theta|\Theta^{(t)}) = \sum_{i=1}^{n}\sum_{k=1}^{K} t_{ik}^{(t)} log(\pi_k f_k(x_i|\theta_k)) \,. \tag{8}$$

M-Step. The M-step updates the current parameters $\Theta^{(t)}$ by maximizing $Q(\Theta|\Theta^{(t)})$ over Θ, so that to have an updated estimate $\Theta^{(t+1)}$. The mixing proportions π_k are computed independently of the component parameters θ_k:

$$\pi_k^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} t_{ik}^{(t)} \,. \tag{9}$$

The update of the parameters θ_k depends on the nature of the mixed components and can be obtained by analyzing the following equation:

$$\sum_{i=1}^{n}\sum_{k=1}^{K} t_{ik}^{(t)} \frac{\partial log f_k(x_i|\theta_k^{(t)})}{\partial\Theta} = 0 \,. \tag{10}$$

3 Belief Function Theory

Before we turn to the EM approach under the TBM framework, we shall sketch out some of the bases of the belief function theory. Details can be found in [11, 14].

3.1 Basic Concepts

Let Ω be a finite non empty set of mutually exclusive elementary events related to a given problem. Ω is generally called the frame of discernment. The set 2^Ω contains all the subsets of Ω: it is the power set of Ω.

The impact of a piece of evidence held by an agent (whatever is it: a sensor, a computer program, an expert, etc) among the propositions of Ω, is expressed by the so-called basic belief assignment (bba). The bba is a function $m^{\Omega} : 2^{\Omega} \rightarrow [0, 1]$ that satisfies: $\sum_{A \subseteq \Omega} m^{\Omega}(A) = 1$.

The value $m^{\Omega}(A)$, called a basic belief mass (bbm), is the quantity of belief that supports exactly the proposition A and that due to the lack of information, does not support any strict subset of A.

The belief function $bel^{\Omega} : 2^{\Omega} \rightarrow [0, 1]$, with $bel^{\Omega}(A) = \sum_{\emptyset \neq B \subseteq A} m^{\Omega}(B)$, expresses the total amount of belief assigned to the subsets implying A without implying $\overline{A}$.

The plausibility function $pl^{\Omega} : 2^{\Omega} \rightarrow [0, 1]$, with $pl^{\Omega}(A) = \sum_{A \cap B \neq \emptyset} m^{\Omega}(B)$, quantifies the degree of belief committed to the propositions compatible with A.

Several special belief functions relative to particular states of uncertainty are defined. The vacuous belief function quantifies a state of total ignorance, in which no support is given to any particular subset of Ω. This function is defined as follows [11]:

$$m^{\Omega}(\Omega) = 1 \text{ and } m^{\Omega}(A) = 0, \forall A \subset \Omega \, . \tag{11}$$

A Bayesian belief function is a belief function where the belief is only allocated among elementary events of Ω [11].

A certain belief function is a Bayesian belief function where the whole belief is assigned to a unique elementary event A: it expresses a state of total certainty. This function is defined by:

$$m^{\Omega}(A) = 1 \text{ and } m^{\Omega}(B) = 0, \forall B \subseteq \Omega, A \in \Omega \text{ and } B \neq A \, . \tag{12}$$

3.2 Combination of Belief Functions

Let $m^{\Omega}_{E_1}$ and $m^{\Omega}_{E_2}$ be two bba's induced from two distinct information sources (E_1 and E_2) and defined on the same frame of discernment Ω. The joint impact of both pieces of evidence is given by the conjunctive rule of combination [12]:

$$(m^{\Omega}_{E_1} \textcircled{\cap} m^{\Omega}_{E_2})(A) = \sum_{B \cap C = A} m^{\Omega}_{E_1}(B) m^{\Omega}_{E_2}(C) \, . \tag{13}$$

3.3 The Pignistic Transformation

In the TBM, beliefs can be held at two levels: a credal level where beliefs are entertained and quantified by belief functions, and a pignistic level, where decisions are made. At this level, beliefs are transformed into probability measures (denoted by $BetP^{\Omega}$) in order to choose the most likely hypothesis. The transformation rule is called the pignistic transformation defined for all $\omega_k \in \Omega$ as:

$$BetP^{\Omega}(w_k) = \sum_{A \ni \omega_k} \frac{m^{\Omega}(A)}{|A|} \frac{1}{(1 - m^{\Omega}(\emptyset))} \, . \tag{14}$$

3.4 The Generalized Bayesian Theorem

Smets [13] has generalized the Bayesian theorem (GBT), offering an interesting tool for inverting conditional belief functions within the TBM framework. Assume that we have a vacuous a priori belief on a frame Ω, and we know for each element $\omega_i \in \Omega$, what would be our beliefs on another frame X if this element happened. Suppose that we learn that the actual value of X is in $x \subseteq X$, then the GBT allows us to derive the conditional belief function over the frame Ω given the observation x. One has:

$$pl^{\Omega}[x](\omega) = 1 - \prod_{\omega_i \in \omega} (1 - pl^{X}[\omega_i](x)) \ . \tag{15}$$

4 The Credal EM Approach

The Credal EM (CrEM) [15] is a variant of EM for partially supervised learning. In this approach, the class label of the observations can be partially known. That is, it can be imprecise and/or uncertain. This knowledge is represented by belief functions as understood in the TBM.

The learning set is then given by: $L = \{(x_1, m_1^{\Omega}), ..., (x_n, m_n^{\Omega})\}$, where $X = \{x_1, ..., x_n\}$ are n iid observations derived from a mixture of K classes $\Omega = \{\omega_1, ..., \omega_K\}$, and $m_i^{\Omega} : 2^{\Omega} \rightarrow [0, 1]$ are the bba's representing the a priori beliefs of membership of the observations x_i into the subsets of Ω.

E-Step. In the classical approach, the algorithm computes the a posteriori probability $t_{ik}^{(t)}$ that x_i has been generated by the class k estimated at the current iteration. The CrEM computes the mass $m^{\Omega}[x_i, \Theta^{(t)}]$ that x_i has been generated by the class k with the current parameters $\Theta^{(t)}$ through the GBT from its corresponding plausibilities:

$$pl^{\Omega}[x_i](A) = 1 - \prod_{\omega_j \in A} (1 - pl^{X}[\omega_j](x_i)) \ , \forall A \subseteq \Omega \ . \tag{16}$$

These masses are then combined with the prior bba's through the conjunctive rule of combination. The resulting masses are given by:

$$\hat{m}^{\Omega}[x_i, \Theta^{(t)}](A) = \sum_{B \cap C = A} m^{\Omega}[x_i, \Theta^{(t)}](B) \ m_i^{\Omega}(C) \ , \forall \ A \subseteq \Omega \ . \tag{17}$$

M-Step. The M-step finds the most probable value of the mixture parameters. This comes down to determine the parameters $\theta \in \Theta$ that maximize the conditional plausibility of the data given θ. Under the iid assumption, this term is given: $\prod_{i=1}^{n} pl^{X}[\theta](x_i)$.

The likelihood function to be maximized is then given by [15]:

$$Q(\Theta|\Theta^{(t)}) = \sum_{i=1}^{n} \sum_{A \subseteq \Omega} \hat{m}^{\Omega}[x_i, \Theta^{(t)}](A) log(pl^{X}[A](x_i)) \ . \tag{18}$$

This equation is analogous to the equation (8) in the TBM framework.

5 Generalized Credal EM

The CrEM provides an alternative for learning in an uncertain environment that is more general than the one proposed in [1] which deals only with imprecise class labels.

However, this approach is not fitted to situations where the values of the attributes characterizing the observations are also partially known. This could involve missing data (some attribute values are missing), imprecise data (we only know that the value of such attributes belongs to a subset of possible values), or uncertain data (we only have some beliefs about the actual value of such attributes).

In this section, we develop a generalization of the CrEM approach that copes with these situations. We first introduce a method that takes into account missing data, then we propose a more general approach that integrates imprecise and uncertain knowledge. Hence, our approach deals with uncertainty in class and attribute values. Besides, we should note that our method deals only with categorical data.

5.1 Learning from Missing Data

In the previous sections, only one aspect of the EM algorithm has been highlighted: learning mixture models. Another important aspect of EM is to learn from data sets with missing values [3, 8]. In this section, we propose to combine this application of EM with that of learning mixture parameters in the TBM framework [15].

We assume that the data X are made up of two components: an observed component X^o and a missing component X^m. Each object x_i in the missing component is divided into (x_i^o, x_i^m) where x_i^o denotes the observed attribute values of x_i and x_i^m the missing attributes, and each x_i can have different missing attributes.

The conditional expected complete data likelihood given the observed data and the current parameters is then written as follows:

$$E[\mathcal{L}_c(\Theta|X^o, X^m, Z)|X^o, \Theta^{(t)}] \; . \tag{19}$$

So, there are two forms of incomplete data: the variables z_{ik} that indicate for each object, which class it comes from, and the missing data x_i^m. The E-step gives an estimation of both forms of missing data: $E[z_{ik}|X^o, \Theta^{(t)}]$ and $E[x_i^m|X^o, \Theta^{(t)}]$. The M-step uses then the completed data to update the mixture model parameters Θ.

E-Step. The first term to be estimated is given by $t_{ik}^{(t)}$, the probability that x_i has been generated from the class k. These probabilities are derived through the pignistic transformation from the masses $\hat{m}^{\Omega}[x_i, \Theta^{(t)}]$:

$$t_{ik}^{(t)} = \sum_{A \ni \omega_k} \frac{\hat{m}^{\Omega}[x_i, \Theta^{(t)}](A)}{|A|} \frac{1}{1 - \hat{m}^{\Omega}[x_i, \Theta^{(t)}](\emptyset)} \; . \tag{20}$$

The masses $\hat{m}^{\Omega}[x_i, \Theta^{(t)}]$ express the beliefs of membership of x_i into the classes of Ω computed as in the CrEM approach (see equation (17)) and measured only over the observed values x_i^o.

The second term gives for each missing attribute x_i^d in x_i^m, the probability that the attribute x_i^d takes the category j (for $j = 1, ..., nb_d$). Since we assume that within each class the attributes are independent, for each class k, we have a different estimation of x_i^{dj} given by $p_k^{dj(t)}$ computed at the current iteration.

M-Step. The M-step updates the current parameters using these expected values. The mixing proportions π_k are updated using the t_{ik} as in equation (9).

The parameters θ_k given by the probabilities p_k^{dj} are updated by:

$$p_k^{dj^{(t+1)}} = \frac{\sum_{i=1}^{n} t_{ik}^{(t)} x_i^{dj}}{\sum_{i=1}^{n} t_{ik}^{(t)}} , \tag{21}$$

where $\sum_{i=1}^{n} t_{ik}^{(t)} x_i^{dj}$ is the estimated number of objects in the class k in which the attribute x_i^d has the category j and $\sum_{i=1}^{n} t_{ik}^{(t)}$ is the total estimated number of objects in the class k. So, $t_{ik} x_i^{dj}$ has to be substituted by $t_{ik} p_k^{dj}$ for the missing components.

5.2 Learning from Partial Knowledge

In this subsection, we propose an approach that integrates imprecise and uncertain knowledge regarding the attribute values characterizing the objects of the learning set. As the prior knowledge about the class labels presented before, this knowledge is represented by belief functions.

The data X, are divided here into two components: a component known with certainty denoted by X^c and an uncertain component X^u. That is, each x_i of X^u is divided into (x_i^c, x_i^u) where x_i^c are the well defined attributes and x_i^u are the partially known attributes.

For uncertain attributes, we use a set of bba's $m_i^{\Omega^d} : 2^{\Omega^d} \rightarrow [0, 1]$ to express the a priori beliefs of the actual value of these attributes. 2^{Ω^d} denotes the power set corresponding to the set of possible values of the attribute d.

Example 1. Let us consider three attributes given by: the salary, the marital status, and the place of residence with respective possible categories:
$\Omega^{salary} = \{low, medium, high\}$,
$\Omega^{mariStat} = \{single, married, divorced, widowed\}$,
$\Omega^{placeRes} = \{apartment, house\}$.

One can have:
$x_1 = (\{low(0.2), medium(0.8)\}, married(1), house(1))$, where the attribute salary is uncertain (with $m_1^{\Omega^{salary}}(low) = 0.2$ and $m_1^{\Omega^{salary}}(medium) = 0.8$) and the remaining attributes are perfectly known.
$x_2 = (\{medium, high\}(1), married(1), \Omega^{placeRes}(1))$, where the attribute salary is imprecise, the attribute marital status is perfectly known and the attribute place of residence is totaly unknown.

This representation is then a generalization of the previous one since it covers the case where attribute values are missing. This is handled through vacuous belief functions. We also notice that the certain case, where all the attributes are perfectly known, can also be modeled here through certain belief functions.

E-Step. The E-step estimates both the variables z_{ik} and the uncertain values of x_i^u. The first values are again given by t_{ik} estimated over the certain component of x_i.

The second values denoted by $E[x_i^u|X^o, \Theta^{(t)}]$ are first estimated using the current parameters $p_k^{dj(t)}$. These probabilities which can be written in the form of Bayesian masses $m_{ik}^{\Omega^{d(t)}}$, are then combined with the a priori masses through the conjunctive rule of combination, to integrate our initial beliefs about the attribute values. The resulting masses denoted by $\hat{m}_{ik}^{\Omega^{d(t)}}$, are given by:

$$\hat{m}_{ik}^{\Omega^{d(t)}}(A) = \sum_{B \cap C = A} m_{ik}^{\Omega^{d(t)}}(B) m_i^{\Omega^d}(C) \ , \ \forall A \subseteq \Omega^d \ . \tag{22}$$

The updated estimation of x_i^{dj} expressing the probability that the attribute x_i^d has the category j, is denoted by $\hat{p}_{ik}^{dj(t)}$. These probabilities are derived from the resulting combined masses using the pignistic transformation:

$$\hat{p}_{ik}^{dj(t)} = \sum_{A \ni \omega_{dj}} \frac{\hat{m}_{ik}^{\Omega^{d(t)}}(A)}{|A|} \frac{1}{(1 - \hat{m}_{ik}^{\Omega^{d(t)}}(\emptyset))} \ , \tag{23}$$

where ω_{dj} denotes the category j of the attribute d.

M-Step. The M-step uses these estimations to update the current parameters as detailed in the previous subsection. The term $t_{ik}\hat{p}_{ik}^{dj}$ is used for the uncertain values in the equation (21).

Note that in both methods, the E and M steps are iterated until the likelihood function $Q(\Theta|\Theta^{(t)}) - Q(\Theta|\Theta^{(t-1)})$ becomes inferior to some threshold ε fixed a priori. This function is given in equation (8). As the classical EM approach [3], the proposed algorithms converge at a stationary point of the mixture parameters and provide a local maximum of the likelihood function.

6 Experimental Results

In order to evaluate our proposed method which consists in a partially supervised EM classification approach with imperfect knowledge at the attribute and class values, we have implemented two algorithms in Matlab V 7.0. Both algorithms deal with uncertain class labels. Besides, the former (GenCrEM_1) handles missing attribute values, whereas the latter (GenCrEM_2) deals with uncertain and/or imprecise attribute values.

We have then applied these algorithms on real databases obtained from the UCI Machine Learning Repository [10]. We have modified these databases in order to disturb their certainty: we have randomly eliminated some attribute values

Table 1. Description of databases

Database	#instances	#attributes	#classes
Balance scale	625	4	3
Wisconsin breast cancer	699	8	2
Car evaluation	1728	6	4

for the GenCrEM_1, and we have randomly introduced bba's in some attribute values for GenCrEM_2 by considering their initial certain values. Moreover, in both cases, we have randomly generated bba's on the class labels by taking into account the initial true labels. In Table 1, a brief description of these databases is given.

We have tested both algorithms for different percentages of missing and uncertain attribute values respectively for the GenCrEM_1 and the GenCrEM_2. We have then applied the CrEM [15] on the certain attribute part of the databases. Table 2 gives the percentages of correctly classified instances (PCC) compared with the initial classification for each database. The mean PCC's obtained from the three methods and measured over the considered databases are given in Figure 1.

It is found that the PCC's produced by GenCrEM_1 are higher than the PCC's given by CrEM for the three databases and for the three considered percentages of imperfect data (20%, 30% and 40%). For instance, in the Car evaluation database and with 40% of imperfect data, the PCC is equal to 70.61% for CrEM and 77.8% for GenCrEM_1. Besides, the results given by GenCrEM_2 are better than the ones given by GenCrEM_1 in all the test cases. For instance, the PCC is equal to 80.36% for GenCrEM_1 and $83,62\%$ for GenCrEM_2 in the Balance scale database and with 30% of imperfect data. So, GenCrEM_2 which is the generalized case, is very appropriate to integrate additional knowledge about the objects of the learning set even if this knowledge is uncertain. Furthermore, it is shown that while the PCC's of our method remain nearly constant and quite high (around 80% for GenCrEM_1 and 83% for GenCrEM_2) when the percentage of imperfect data increases, the PCC of CrEM shows considerable decrease from 75.3% (for 20% of imperfect data) to 69.89% (for 40% of imperfect data). So,

Table 2. Experimental results

	Balance scale			Wisconsin b.c.			Car evaluation		
percent_imperf_obj	20%	30%	40%	20%	30%	40%	20%	30%	40%
CrEM (in %)	75.25	71.67	69.5	73.87	70.33	69.56	76.8	73.23	70.61
GenCrEM_1 (in %)	80.67	80.36	80.62	81.26	82.5	82.69	78.73	78.26	77.8
GenCrEM_2 (in %)	82.4	83.62	82.85	84.53	83.82	84.68	82.13	81.75	82.28

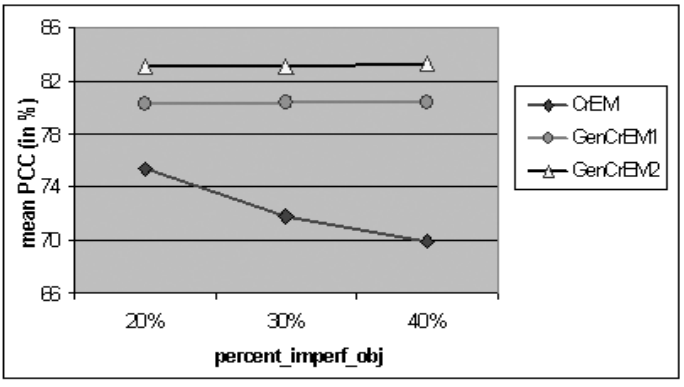


Fig. 1. Experimental results

our proposed approach is more appropriate to handle partially known attribute values.

We should mention that with our proposed method and if all the attribute bba's are certain, the results are equivalent to the CrEM. Besides, if both attribute and class values are perfectly known, that is when we are in a state of total certainty, the results are analogous to those obtained from the classical EM algorithm [3]. Note that when the class labels are imprecise, the CrEM produces very similar results than [1]. So our proposed approach is a generalization of these methods.

7 Conclusion

In this paper, we have proposed an EM approach for learning in an uncertain environment. The uncertainty is represented by belief functions as understood in the TBM. This approach is adapted for cases where not only the knowledge about the classes of the objects can be partial but also their characteristics. Our method provides a more flexible tool to deal with these situations. Future works are concerned with both continuous and mixed data. We will also focus on the model selection issue which notably includes the choice of the mixture components.

References

[1] Ambroise, C., Govaert, G.: EM algorithm for partially known labels. In: IFCS'2000, Namur, Belgium, vol. 1, pp. 161–166 (2000)

[2] Ben Hariz, S., Elouedi, Z., Mellouli, K.: Clustering approach using belief function theory. In: Euzenat, J., Domingue, J. (eds.) AIMSA 2006. LNCS (LNAI), vol. 4183, pp. 162–171. Springer, Heidelberg (2006)

[3] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, series B 39, 1–38 (1977)
[4] Denoeux, T.: A k-nearest neighbor classification rule based on Dempster-Shafer theory. IEEE Transactions on Systems, Man and Cybernetics 25(5), 804–813 (1995)
[5] Elouedi, Z., Mellouli, K., Smets, P.: Belief decision trees: theoretical foundations. International Journal of Approximate Reasoning 28, 91–124 (2001)
[6] Everitt, B.: An introduction to latent variable models. Chapman and Hall, Sydney (1984)
[7] Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of finite mixture models. IEEE Transaction on Pattern Analysis and Machine Intelligence 24(3), 381–396 (2002)
[8] Ghahramani, Z., Jordan, M.I.: Supervised learning from incomplete data via an EM approach. In: Cowan, J.D., Tesauro, G., Alspector, J. (eds.) Advances in Neural Information Processing Systems, vol. 6, pp. 120–127. Morgan Kaufmann Publishers, San Francisco (1994)
[9] McLachlan, G.J., Basford, K.E.: Mixture models. Inference and application to clustering. Marcel Dekker, New York (1989)
[10] Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
[11] Shafer, G.: A Mathematical Theory of Evidence. Princeton Univ. Press, Princeton (1976)
[12] Smets, P.: The combination of evidence in the transferable belief model. IEEE Pattern analysis and Machine Intelligence 12(5), 447–458 (1990)
[13] Smets, P.: Belief functions: The disjunctive rule of combination and the generalized Bayesian theorem. International Journal of Approximate Reasoning 9, 1–35 (1993)
[14] Smets, P., Kennes, R.: The transferable belief model. Artificial Intelligence 66, 191–234 (1994)
[15] Vannoorenberghe, P., Smets, P.: Partially supervised learning by a credal EM approach. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 956–967. Springer, Heidelberg (2005)

Logical Compilation of Bayesian Networks with Discrete Variables

Michael Wachter[1] and Rolf Haenni[1,2]

[1] University of Bern, Switzerland
{wachter,haenni}@iam.unibe.ch
[2] Bern University of Applied Sciences, Switzerland
rolf.haenni@bfh.ch

Abstract. This paper presents a new direction in the area of compiling Bayesian networks. The principal idea is to encode the network by logical sentences and to compile the resulting encoding into an appropriate form. From there, all possible queries are answerable in linear time relative to the size of the logical form. Therefore, our approach is a potential solution for real-time applications of probabilistic inference with limited computational resources. The underlying idea is similar to both the differential and the weighted model counting approach to inference in Bayesian networks, but at the core of the proposed encoding we avoid the transformation from discrete to binary variables. This alternative encoding enables a more natural solution.

1 Introduction

As Bayesian networks (BN) are more and more applied to complex real-world applications, the development of fast and flexible inference methods becomes increasingly important. In the last decades, researchers have developed various kinds of exact and approximate inference algorithms, each of them with corresponding advantages and disadvantages. Some methods are particularly designed for real-time inference with limited computational resources such as time or memory. See [1] for a comprehensive and compact survey.

Two particular inference methods are the weighted model counting [2] and the differential approach [3,4]. The former suggests to view a BN as a CNF model counting problem. It encodes the given BN as a CNF, and employs techniques used in the state-of-the-art SAT and model counting engines to solve the problem. The differential approach suggests to view a BN as a *multi-linear function* (MLF), the so-called *network polynomial*, from which answers to probabilistic queries are retrieved by differentiating the polynomial. Relative to the given BN, the network polynomial is exponential in size, but it is possible to efficiently encode it by a CNF of linear size. As suggested in [5], this CNF is then compiled into a *decomposable negation normal form* (DNNF) with the additional properties of *smoothness* and *determinism* [6]. The resulting sd-DNNF is an intermediate step, from which an arithmetic circuit is extracted, whose size is not necessarily exponential relative to the original BN. This arithmetic circuit

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 536–547, 2007.

is guaranteed to compute the original network polynomial, and can therefore be used to obtain all necessary partial derivatives in time (and space) linear to its size. In its essence, the aim of the whole procedure is to generate a preferably optimal factoring of the network polynomial.

Such a logical approach is beneficial in many ways. The most important advantage is the ability to encode *context-specific independences* or other local regularities in the given conditional probability tables [4,7]. This inherently includes appropriate solutions for the particular type of CPT obtained from noisy-OR and noisy-AND nodes or from purely logical relations. In comparison with classical inference methods such as join-tree propagation or message-passing, which do not directly exploit such local structures, there has been reports of tremendous improvements in both compile time and online inference [8,9]. Another advantage is the ability to efficiently update numerical computations with minimal computational overhead. This is a key prerequisite for experimental sensitivity analyses.

1.1 Overview of the Method

In [10], we showed how to use the CNF encoding from [2] as a starting point for a compilation in the sense of [3,5], but our analysis was restricted to binary variables only. In this paper, we will break out of the binary case and discuss the compilation of BNs with (finite) discrete variables. For this, a discrete variable $\Theta_{X|\mathbf{y}}$ is attributed to each conditional probability distribution $P(X|\mathbf{y})$ in the CPT of a network variable X with a finite set of states Ω_X, e.g. $\Omega_X = \{high, medium, low\}$. As a result, the generated logical representation ψ consists of two types of variables, the ones linked to the CPT entries and the network variables. The corresponding sets of variables are denoted by Θ and Δ, respectively.

In order to use the logical representation ψ to compute the posterior probability $P(\mathbf{q}|\mathbf{e}) = P(\mathbf{q},\mathbf{e})/P(\mathbf{e})$ of a query event $\mathbf{q} \in \Omega_{\mathbf{Q}} = \Omega_{Q_1} \times \cdots \times \Omega_{Q_r}$ given the evidence $\mathbf{e} \in \Omega_{\mathbf{E}} = \Omega_{E_1} \times \cdots \times \Omega_{E_s}$, it is sufficient to look at the simpler problem of computing prior probabilities $P(\mathbf{x})$ of arbitrary conjunctions $\mathbf{x} \in \Omega_{\mathbf{X}} = \Omega_{X_1} \times \cdots \times \Omega_{X_t}$ in order to obtain corresponding numerators $P(\mathbf{q},\mathbf{e})$ and denominators $P(\mathbf{e})$. Our solution for this consists of the following three steps:

1. Condition ψ on $\mathbf{x} \in \Omega_{\mathbf{X}}$ to obtain $\psi|\mathbf{x}$.
2. Eliminate (forget) from $\psi|\mathbf{x}$ the variables Δ. The resulting logical representation of $[\psi|\mathbf{x}]^{-\Delta}$ consists of variables from Θ only.
3. Compute the probability of the event represented by $[\psi|\mathbf{x}]^{-\Delta}$ to obtain $P(\mathbf{x}) = P([\psi|\mathbf{x}]^{-\Delta})$. For this, we assume that the variables $\Theta_{X|\mathbf{y}} \in \Theta$ are probabilistically independent and that $P(\Theta_{X|\mathbf{y}}{=}\theta_{x|\mathbf{y}}) = P(X{=}x|\mathbf{y})$ are the respective marginal probabilities for $\theta_{x|\mathbf{y}} \in \Omega_{\Theta_{X|\mathbf{y}}}$ and $x \in \Omega_X$.

For the choice of an appropriate target compilation language for ψ, it is thus necessary to select a language that supports two transformations (conditioning

Table 1. Main differences between [2], [5], [10] and the paper at hand

Approach	[2]	[5]	[10]	paper at hand
Encoding Type	1	2	1	1
Multi-state Variables	transformation to binary variables	transformation to binary variables	not allowed	no transformation
Logical Compilation	no	yes	yes	yes

and forgetting) and one query (probability computation) in polynomial time. At first sight, just by looking at the results given in [11], it seems that no such language exists. However, as we will see in this paper, we can exploit the fact that the variables in Δ satisfy a certain property w.r.t. ψ. The particular form of forgetting such *deterministic* variables is called *deterministic forgetting*, and we shall see that it is (at least) supported by two languages. However, since probability computations are only supported by one of them our search for an appropriate target compilation language for BNs thus leads to *multi-state directed acyclic graphs* (MDAG) which satisfy the properties of *decomposability*, *determinism*, and *no-negations*, denoted as cdn-MDAG. This language is the only representation language that supports all necessary operations of the above procedure in polynomial time. The suggested use of cdn-MDAGs has an obvious advantage over the existing compilation methods: There is no need to replace discrete variables with $\ell > 2$ states by $\ell - 1$ (or ℓ) binary variables. Both the model counting and the differential approach are based on this replacement.

1.2 Contribution and Outline

The conclusion that cdn-MDAGs (the multi-state version of d-DNNFs) should be used as target compilation language for BNs confirms Darwiche's precursory work in [5], but it also shows that Darwiche's additional requirement of smoothness is not really needed. While the actual reasons for this conclusion and the exact role of smoothness remain rather nebulous in [5], a precise and conclusive explanation in terms of the (extended) knowledge compilation map is given in this paper.

Compared with previous works on representing BNs by logical formulae, our approach differs from [2] in the handling of multi-state variables, and the fact that [2] does not at all talk about logical compilation. Yet, the encoding is the same in the case of binary variables. Compared with [5], the encoding is already different in the binary case (see detailed discussion in Section 4 of [10]). In addition, we do not transform the multi-state variables into binary ones. The Table 1 summarizes the main differences between [2], [5], [10] and the paper at hand.

The proposed encoding or this paper enables a more direct computational procedure in terms of a few basic operations of the knowledge compilation map. In our opinion, this is a significant simplification over Darwiche's original method of viewing posterior probabilities as partial derivatives of multi-linear functions,

from which the rather cumbersome process of transforming the CNF encoding via a smooth d-DNNF to an arithmetic circuit (with all negative literals set to 1) results. In the light of this paper, some steps of this process appear as an unnecessary detour, e.g., the transformation of multi-state variables into binary variables or the requirement of smoothness. In a nutshell, we believe that the method of this paper is an important contribution to the area of compiling BNs, mainly as a significant advancement in terms of clarity and simplicity.

The structure of this paper is as follows. Section 2 provides a short summary of possible representations of *Cartesian indicator functions* (CIF) and the corresponding knowledge compilation map. Then, we formalize the concepts of deterministic variables and deterministic forgetting, and extend the knowledge compilation map accordingly. The topic of Section 3 is the logical representation and evaluation of BNs. This part includes the main theorems of the paper. Section 4 concludes the paper.

2 Representing Cartesian Indicator Functions

Let $\mathbf{V} = \{V_1, \ldots, V_v\}$ be a set of v variables and suppose that Ω_{V_i} denotes the finite set of states of V_i. A finite indicator function f is defined by $f : \Omega_{\mathbf{V}} \to \mathbb{B}$, where $\Omega_{\mathbf{V}} = \Omega_{V_1} \times \cdots \times \Omega_{V_v}$ and $\mathbb{B} = \{0, 1\}$. To emphasize the fact that f is a mapping from the Cartesian product $\Omega_{V_1} \times \cdots \times \Omega_{V_v}$ to $\mathbb{B}$, f is called a *Cartesian indicator function* (CIF). The so-called *satisfying set* $S_f = \{\mathbf{x} \in \Omega_{\mathbf{V}} : f(\mathbf{x}) = 1\} = f^{-1}(1)$ of f is the set of v-dimensional vectors $\mathbf{x} \in \Omega_{\mathbf{V}}$ for which f evaluates to 1. Composed CIFs will be specified using the logical connectors $\wedge$, $\vee$, $\neg$, $\rightarrow$, and $\leftrightarrow$ in their usual interpretation. Special cases of finite CIFs are Boolean functions (BF), where $\Omega_{V_i} = \mathbb{B}$, and therefore $\Omega_{\mathbf{V}} = \mathbb{B}^v$.

2.1 Representation Languages

To start with the least restrictive view w.r.t. possible representation languages, consider the concept of a *multi-state* DAG (or MDAG for short). According to [11], MDAGs are rooted, directed, acyclic graphs, in which each leaf node is represented by $\Box$ and labeled with $\top$ (true), $\bot$ (false), or $X{=}x$, where $X \in \mathbf{V}$ is a variable, and $x \in \Omega_X$ is one of its states. Each internal node is represented by $\triangle$ (logical and), $\triangledown$ (logical or), or $\Diamond$ (logical not). The set of all possible MDAGs of $\mathbf{V}$ is called *language* and denoted by $\texttt{MDAG}_{\mathbf{V}}$ or simply $\texttt{MDAG}$. In a MDAG, each node α represents a finite CIF f_α by

$$f_\alpha = \begin{cases} \wedge_{i=1}^{t} f_{\beta_i}, & \text{if } \alpha \text{ is an } \triangle\text{-node with children } \beta_1, \ldots, \beta_t, \\ \vee_{i=1}^{t} f_{\beta_i}, & \text{if } \alpha \text{ is an } \triangledown\text{-node with children } \beta_1, \ldots, \beta_t, \\ \neg f_\psi, & \text{if } \alpha \text{ is a } \Diamond\text{-node with the child } \psi, \\ 1, & \text{if } \alpha \text{ is a } \Box\text{-node labeled with } \top, \\ 0, & \text{if } \alpha \text{ is a } \Box\text{-node labeled with } \bot, \\ f_{X=x}, & \text{if } \alpha \text{ is a } \Box\text{-node labeled with } X{=}x, \end{cases}$$

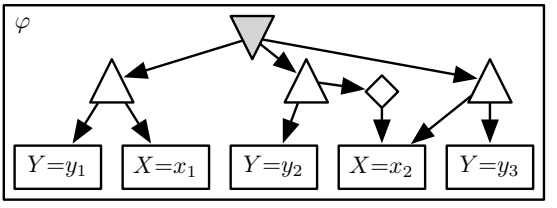


Fig. 1. The finite CIF f represented as the MDAG φ

where $f_{X=x}(\mathbf{x})$ with $\mathbf{x} \in \Omega_{\mathbf{V}}$ is defined by

$$f_{X=x}(\mathbf{x}) = \begin{cases} 1, & \text{if } x \text{ is the corresponding value of } X \text{ in } \mathbf{x}, \\ 0, & \text{otherwise.} \end{cases}$$

The MDAG depicted in Fig. 1 represents the finite CIF $f = ([Y{=}y_1] \wedge [X{=}x_1]) \vee ([Y{=}y_2] \wedge \neg[X{=}x_2]) \vee ([X{=}x_2] \wedge [Y{=}y_3])$.

Our convention is to denote MDAGs by lower-case Greek letters such as φ, ψ, or the like. Two MDAGs φ, ψ are *equivalent*, denoted by $\varphi \equiv \psi$, iff $f_\varphi = f_\psi$. Furthermore, φ *entails* ψ, denoted by $\varphi \models \psi$, iff $f_\varphi(\mathbf{x}) \leq f_\psi(\mathbf{x})$ for all $\mathbf{x} \in \Omega$. The set of variables included in $\varphi \in$ `MDAG` is denoted by $vars(\varphi) \subseteq \mathbf{V}$. The number of edges of φ is called its *size* and is denoted by $|\varphi|$. MDAGs may satisfy various properties [11], but in the context of this paper, only three of them are relevant:

- *Decomposability* (c): the sets of variables of the children of each $\triangle$-node α in φ are pairwise disjoint (i.e. if $\beta_1, \ldots, \beta_n$ are the children of α, then $vars(\beta_i) \cap vars(\beta_j) = \emptyset$ for all $i \neq j$);
- *Determinism* (d): the children of each $\triangledown$-node α in φ are pairwise logically contradictory (i.e. if $\beta_1, \ldots, \beta_n$ are the children of α, then $\beta_i \wedge \beta_j \equiv \bot$ for all $i \neq j$);
- *No-Negation* (n):[1] φ does not contain any $\Diamond$-node.

A de**c**omposable and **d**eterministic MDAG is called cd-MDAG, and `cd-MDAG` refers to the corresponding language, a sub-language of `MDAG`. The example shown in Fig. 1 is a cd-MDAG.

Another important sub-language is `cn-MDAG`. It refers to the sub-language of `MDAG` where decomposability and no-negation are satisfied. `cdn-MDAG` is the sub-language of `cn-MDAG`, where determinism is satisfied in addition to decomposability and no-negation. This language includes Darwiche's `d-DNNF` language as a special case, when all variables are binary. Other sub-languages are obtained from considering further properties, e.g. `OMDD` (ordered multivalued decision diagram) is the sub-language of `cdn-MDAG` satisfying *decision*, *read-once*, and *ordering*. For a more comprehensive overview and a detailed discussion we refer to [11].

[1] *No-negation* corresponds to *simple-negation* in [12,13].

Table 2. Sub-languages of the MDAG language and their supported queries and transformations. $\surd$ means "supports", $\bullet$ means "does not support", $\circ$ means "does not support unless $P = NP$", and ? means "unknown".

	CO/CE	VA/IM	CT/PR/PEQ	EQ	SE	TC	FO	SFO	AND	AND_2	OR	OR_2	NOT
MDAG	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\surd$	$\circ$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
cn-MDAG	$\surd$	$\circ$	$\circ$	$\circ$	$\circ$	$\surd$	$\surd$	$\surd$	$\circ$	$\circ$	$\surd$	$\surd$	$\circ$
cd-MDAG	$\surd$	$\surd$	$\surd$	?	$\circ$	$\surd$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\surd$
cdn-MDAG	$\surd$	$\surd$	$\surd$	?	$\circ$	$\surd$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	?
OMDD	$\surd$	$\surd$	$\surd$	$\surd$	$\circ$	$\surd$	$\bullet$	$\surd$	$\bullet$	$\circ$	$\bullet$	$\circ$	$\surd$

2.2 Succinctness, Queries, and Transformations

A language $\mathtt{L}_1$ is *equally or more succinct* than another language $\mathtt{L}_2$, denoted by $\mathtt{L}_1 \preceq \mathtt{L}_2$, if any sentence $\alpha_2 \in \mathtt{L}_2$ has an equivalent sentence $\alpha_1 \in \mathtt{L}_1$ whose size is polynomial in the size of α_2. A language $\mathtt{L}_1$ is *strictly* more succinct than another language $\mathtt{L}_2$, denoted by $\mathtt{L}_1 \prec \mathtt{L}_2$, iff $\mathtt{L}_1 \preceq \mathtt{L}_2$ and $\mathtt{L}_2 \not\preceq \mathtt{L}_1$. With respect to the above-mentioned languages, we have the following proven relationships [11]:

$$\text{MDAG} \prec \left\{ \begin{array}{l} \text{cn-MDAG} \prec \\ \text{cd-MDAG} \preceq \end{array} \right\} \text{cdn-MDAG} \prec \text{OMDD}.$$

It is still unknown whether cd-MDAG is strictly more succinct than cdn-MDAG or not.

Queries are operations that return information about a finite CIF without changing its MDAG representation. The most important queries are *consistency* (CO), *validity* (VA), *clause entailment* (CE), *term implication* (IM), *sentential entailment* (SE), *equivalence* (EQ), *model counting* (CT), *probabilistic equivalence* (PEQ), and *probability computation* (PR).

Finally, a *transformation* is an operation that returns a MDAG representing a modified finite CIF. The new MDAG is supposed to satisfy the same properties as the language in use. The most important transformations are (*term*) *conditioning* (TC), *forgetting* (FO), *singleton forgetting* (SFO), *general/binary conjunction* (AND/AND_2), *general/binary disjunction* (OR/OR_2), and *negation* (NOT).

If a language supports a query or transformation in polynomial time with respect to the size of the involved MDAGs, we say that it *supports* this query or transformation. Table 2 shows the supported queries and transformations of the considered languages [11].

2.3 Deterministic Variables

It is interesting to see in Table 2 that forgetting is supported by cn-MDAG but not by cdn-MDAG or cd-MDAG. This is a consequence of the fact that forgetting does not preserve determinism in general. Let us now have a look at the particular case of variables which preserve determinism while being forgotten, i.e. we formalize the idea of *determined* variables as introduced in [14].

Definition 1. *For $\varphi \in \texttt{MDAG}$, the variable $X \in \mathbf{V}$ is called* deterministic *w.r.t. φ, denoted by $X \,||\, \varphi$, iff $[\varphi|x] \wedge [\varphi|x'] \equiv \bot$ for all states $x, x' \in \Omega_X$, $x \neq x'$.*

The process of forgetting deterministic variables will be discussed in the next subsection. Before, let's have a look at some basic properties of deterministic variables. In the following, let X be a variable with $\Omega_X = \{x_1, \ldots, x_m\}$.

Theorem 1. *$X \,||\, \varphi$ implies $X \,||\, \psi$ for all $\psi \in \texttt{MDAG}$ with $\psi \models \varphi$.*

Theorem 2. *Let $\varphi = \bigwedge_{i=1}^{m}((X{=}x_i) \leftrightarrow \varphi_i)$ such that $\varphi_i \wedge \varphi_j \equiv \bot$ and $X \notin vars(\varphi_i)$ for all $i, j \in \{1, \ldots, m\}$. This implies $X \,||\, \varphi$.*

The proofs are analog to the proofs of the corresponding theorems in [10]. An immediate consequence is the following corollary, which is necessary to prove one of the main theorems of Section 3.

Corollary 1. *Let $\varphi = \bigwedge_{i=1}^{m}((X{=}x_i) \leftrightarrow \varphi_i)$ such that $\varphi_i \wedge \varphi_j \equiv \bot$ and $X \notin vars(\varphi_i)$ for all $i, j \in \{1, \ldots, m\}$. This implies $X \,||\, \varphi \wedge \psi$ for all $\psi \in \texttt{MDAG}$.*

For the forgetting of more than one variable, it is useful to generalize the definition of a single deterministic variable to sets of deterministic variables.

Definition 2. *For $\varphi \in \texttt{MDAG}$, the set of variables $\mathbf{X} = \{X_1, \ldots, X_n\} \subseteq \mathbf{V}$ is called* deterministic *w.r.t φ, denoted by $\mathbf{X} \,||\, \varphi$ or simply $X_1, \ldots, X_n \,||\, \varphi$, iff $[\varphi|\mathbf{x}] \wedge [\varphi|\mathbf{x}'] \equiv \bot$ for all instantiations $\mathbf{x}, \mathbf{x}' \in \Omega_{\mathbf{X}}$, $\mathbf{x} \neq \mathbf{x}'$.*

Note that $X, Y \,||\, \varphi$ implies $X \,||\, \varphi$ and $Y \,||\, \varphi$, while the converse is not always true.

2.4 Deterministic Forgetting

Let $\mathbf{W} \subseteq \mathbf{V}$ be a subset of variables, $X \in \mathbf{V}$ a single variable, and φ an arbitrary MDAG. Forgetting the variables $\mathbf{W}$ from φ generates a new MDAG $\varphi^{-\mathbf{W}}$, in which the variables from $\mathbf{W}$ are no longer included, and such that its satisfying set $S_{\varphi^{-\mathbf{W}}}$ is the projection of S_φ to the restricted set of variables $\mathbf{V} \setminus \mathbf{W}$. In the literature, forgetting was originally called *elimination of middle terms* [15], but it is also common to call it *projection*, *variable elimination*, or *marginalization* [16]. There is also a one-to-one analogy to the elimination of existential quantifiers in *quantified (Boolean) formulas* [17], as discussed below.

Singleton forgetting is forgetting with $\mathbf{W} = \{X\}$. A general and simple way to realize singleton forgetting is by constructing a MDAG of the form

$$\varphi^{-X} = \bigvee_{x \in \Omega_X} [\varphi|x].$$

Note that if X is binary, φ^{-X} is logically equivalent to the quantified Boolean formula $(\exists x)\varphi$. It is easy to see that singleton forgetting preserves the properties of simple-negation and decomposability (if present), while determinism is not

preserved (the children of the new ∇-node are not necessarily logically contradictory). This is the reason why singleton forgetting is only supported by MDAG and cn-MDAG, but not by cd-MDAG or cdn-MDAG (see Table 2).

Forgetting multiple variable is usually realized as a sequence of singleton forgetting. In general, this may result in an exponential blow-up of the MDAG size, but the decomposability of cn-MDAG allows to keep this blow-up under control. This is the reason why cn-MDAG is the only language to support forgetting in general. For the details of a corresponding algorithm (for binary variables), we refer to [18].

Now let's turn our attention to the special case of forgetting deterministic variables. One way to look at it is to define two additional transformations called *deterministic forgetting* ($\mathtt{FO_d}$) and *deterministic singleton forgetting* ($\mathtt{SFO_d}$). They correspond to FO and SFO, respectively, but are only applicable to deterministic variables.

For $\mathbf{X} \,||\, \varphi$, the children of the new ∇-node of $\bigvee_{\mathbf{x} \in \Omega_{\mathbf{X}}} [\varphi|\mathbf{x}]$ are logically contradictory by definition. In other words, forgetting deterministic variables preserves determinism. Thus, we can use the forgetting algorithm of cn-MDAG for forgetting deterministic variables in the context of cdn-MDAG. As a consequence, $\mathtt{SFO_d}$ and $\mathtt{FO_d}$ are both supported by cdn-MDAG, as stated in the following theorem.

Theorem 3.

a) MDAG *supports* $\mathtt{SFO_d}$*, but it does not support* $\mathtt{FO_d}$ *unless* $P = NP$.
b) cn-MDAG *and* cdn-MDAG *support* $\mathtt{FO_d}$ *and* $\mathtt{SFO_d}$.
c) cd-MDAG *and* OMDD *support* $\mathtt{SFO_d}$.

The proof is analog to the one given in [10]. Whether cd-MDAG and OMDD support $\mathtt{FO_d}$ is an open question.

3 Compiling Bayesian Networks

The goal of this section is to show that the probability distribution induced by a BN can be represented by a so-called *multi-state CNF* (MCNF) [11] (1st subsection) and that the cdn-MDAG compilation of this MCNF can be used to efficiently compute arbitrary posterior probabilities (2nd subsection). The proposed MCNF representation is similar but not equivalent to the one proposed by Darwiche in [5], for details consider [10]. However, if all variables of the network are binary, the MCNF is equivalent to the CNF proposed in [2].

A *Bayesian network* (BN) is a compact graphical model of a complex probability distribution over a set of variables $\Delta = \{X_1, \ldots, X_n\}$ [19]. It consists of two parts: a DAG representing the direct influences among the variables, and a set of conditional probability tables (CPT) quantifying the strengths of these influences. The whole BN represents the exponentially sized *joint probability distribution* over its variables in a compact manner by

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|parents(X_i)),$$

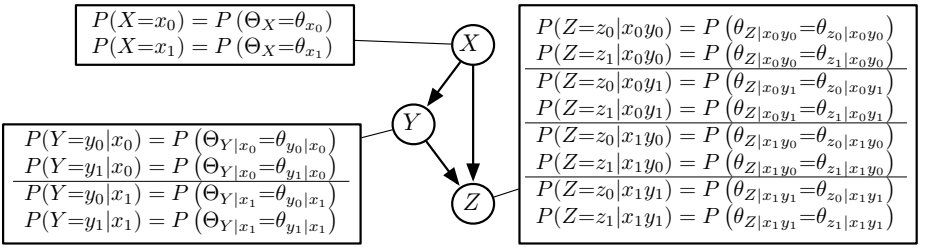


Fig. 2. Example of a Bayesian network

where $parents(X_i)$ denotes the parents of node X_i in the DAG. Figure 2 depicts a small BN with three variables X, Y, and Z.

3.1 Logical Representation

Consider a variable $X \in \Delta$ with $parents(X) = \{Y_1, \ldots, Y_u\} = \mathbf{Y}$, $\Omega_X = \{x_1, \ldots, x_t\}$, and the corresponding CPT. Since X has u parents, the CPT will have $|\Omega_{\mathbf{Y}}| \geq 2^u$ conditional probability distributions, i.e. one conditional probability distribution $P(X|\mathbf{y})$ for each instantiation $\mathbf{y} \in \Omega_{\mathbf{Y}}$ of $\mathbf{Y}$. For each $P(X|\mathbf{y})$, we introduce an auxiliary variable $\Theta_{X|\mathbf{y}}$ with $\Omega_{\Theta_{X|\mathbf{y}}} = \{\theta_{x_1|\mathbf{y}}, \ldots, \theta_{x_t|\mathbf{y}}\}$. Assuming that the variables $\Theta_{X|\mathbf{y}}$ represent probabilistically independent events, we define their respective marginal probabilities by $P(\Theta_{X|\mathbf{y}}{=}\theta_{x|\mathbf{y}}) = P(X{=}x|\mathbf{y})$, as shown in Fig. 2.

To see how the proposed logical representation of the BN works, take a closer look at one particular instantiation $\mathbf{y}$ of $parents(X)$. The idea is that if $\mathbf{y}$ happens to be the true state of $parents(X)$, then $\Theta_{X|\mathbf{y}}{=}\theta_{x|\mathbf{y}}$ is supposed to logically imply $X{=}x$. This logical relationship between $Y_1{=}y_1, \ldots, Y_u{=}y_u$, $\Theta_{X|\mathbf{y}}{=}\theta_{x|\mathbf{y}}$ with $\mathbf{y} = y_1 \cdots y_u \in \Omega_{\mathbf{Y}}$ and $X{=}x$ is expressed by the implication in the following logical expression. By taking the conjunction of all such implications over all instantiations $\mathbf{y}$, we obtain a logical representation ψ_X of the node X with its relationship to its parents:

$$\psi_X = \bigwedge_{\substack{\mathbf{y} \in \Omega_{\mathbf{Y}} \\ \mathbf{y}=y_1 \cdots y_u}} \left(\bigwedge_{\theta_{x|\mathbf{y}} \in \Omega_{\Theta_{X|\mathbf{y}}}} \Big([Y_1{=}y_1] \wedge \cdots \wedge [Y_u{=}y_u] \wedge [\Theta_{X|\mathbf{y}}{=}\theta_{x|\mathbf{y}}] \rightarrow [X{=}x] \Big) \right)$$

A logical representation ψ_Δ of the whole BN is the conjunction

$$\psi_\Delta = \bigwedge_{X \in \Delta} \psi_X$$

over all network variables $X \in \Delta$. This sentence includes two types of variables, the ones linked to the CPT entries and the network variables. The respective sets

of variables are denoted by Θ and Δ, respectively.[2] Note that ψ_X and therewith ψ_Δ is a MCNF, as each of its implications can be written as a clause. For the BN of Fig. 2, we have $\Theta = \{\Theta_X, \Theta_{Y|x_0}, \Theta_{Y|x_1}, \Theta_{Z|x_0y_0}, \Theta_{Z|x_0y_1}, \Theta_{Z|x_1y_0}, \Theta_{Z|x_1y_1}\}$, $\Delta = \{X, Y, Z\}$, and

$$\psi_\Delta = \bigwedge \left(\begin{array}{l} \left.\begin{array}{l} [\Theta_X{=}\theta_{x_0}]{\rightarrow}[X{=}x_0] \\ [\Theta_X{=}\theta_{x_1}]{\rightarrow}[X{=}x_1] \end{array}\right\} \text{ from } \psi_X \\ \left.\begin{array}{l} [X{=}x_0]{\wedge}[\Theta_{Y|x_0}{=}\theta_{y_0|x_0}]{\rightarrow}[Y{=}y_0] \\ [X{=}x_0]{\wedge}[\Theta_{Y|x_0}{=}\theta_{y_1|x_0}]{\rightarrow}[Y{=}y_1] \\ [X{=}x_1]{\wedge}[\Theta_{Y|x_1}{=}\theta_{y_0|x_1}]{\rightarrow}[Y{=}y_0] \\ [X{=}x_1]{\wedge}[\Theta_{Y|x_1}{=}\theta_{y_1|x_1}]{\rightarrow}[Y{=}y_1] \end{array}\right\} \text{ from } \psi_Y \\ \left.\begin{array}{l} [X{=}x_0]{\wedge}[Y{=}y_0]{\wedge}[\Theta_{Z|x_0y_0}{=}\theta_{z_0|x_0y_0}]{\rightarrow}[Z{=}z_0] \\ [X{=}x_0]{\wedge}[Y{=}y_0]{\wedge}[\Theta_{Z|x_0y_0}{=}\theta_{z_1|x_0y_0}]{\rightarrow}[Z{=}z_1] \\ [X{=}x_0]{\wedge}[Y{=}y_1]{\wedge}[\Theta_{Z|x_0y_1}{=}\theta_{z_0|x_0y_1}]{\rightarrow}[Z{=}z_0] \\ [X{=}x_0]{\wedge}[Y{=}y_1]{\wedge}[\Theta_{Z|x_0y_1}{=}\theta_{z_1|x_0y_1}]{\rightarrow}[Z{=}z_1] \\ [X{=}x_1]{\wedge}[Y{=}y_0]{\wedge}[\Theta_{Z|x_1y_0}{=}\theta_{z_0|x_1y_0}]{\rightarrow}[Z{=}z_0] \\ [X{=}x_1]{\wedge}[Y{=}y_0]{\wedge}[\Theta_{Z|x_1y_0}{=}\theta_{z_1|x_1y_0}]{\rightarrow}[Z{=}z_1] \\ [X{=}x_1]{\wedge}[Y{=}y_1]{\wedge}[\Theta_{Z|x_1y_1}{=}\theta_{z_0|x_1y_1}]{\rightarrow}[Z{=}z_0] \\ [X{=}x_1]{\wedge}[Y{=}y_1]{\wedge}[\Theta_{Z|x_1y_1}{=}\theta_{z_1|x_1y_1}]{\rightarrow}[Z{=}z_1] \end{array}\right\} \text{ from } \psi_Z \end{array} \right) .$$

3.2 Computing Posterior Probabilities

The goal of a BN is the computation of the posterior probability $P(\mathbf{q}|\mathbf{e}) = P(\mathbf{q}, \mathbf{e})/P(\mathbf{e})$ of a query event $\mathbf{q} \in \Omega_\mathbf{Q}$ given the observed evidence $\mathbf{e} \in \Omega_\mathbf{E}$. As mentioned in Section 1, it is sufficient to look at the simpler problem of computing prior probabilities $P(\mathbf{x})$ of arbitrary conjunctions $\mathbf{x} \in \Omega_\mathbf{X}$. The following theorem states that the essential step to solve this problem is to forget the propositions Δ from ψ_Δ (or any equivalent form of it) conditioned on $\mathbf{x}$.

[2] The representation of a BN by a logical sentence ψ_Δ over two sets of variables Θ and Δ, together with the given marginal probabilities for the variables in Θ and the corresponding independence assumptions, puts this approach in the broader context of *probabilistic argumentation* [20,21]. This is a theory of formal reasoning which aims at unifying the classical fields of logical and probabilistic reasoning. The principal idea is to evaluate the credibility of a hypothesis by non-additive *probabilities of provability* (or *degrees of support*). This is a natural extension of the classical concepts of probability (in probability theory) and provability (in logic) [20]. The non-additivity of this measure is an important characteristic to distinguish properly between uncertainty and ignorance, but the particularity of the model in this paper always causes the resulting probabilities of provability to degenerate into ordinary (additive) probabilities. The embedding into the theory of probabilistic argumentation has no practical significance for the method and goals of this paper, but it allows inference in BNs to be seen from a totally new perspective. We expect this perspective to be useful as a starting point to study inference in BNs with missing parameters.

Theorem 4. $P(\mathbf{x}) = P\left([\psi_\Delta|\mathbf{x}]^{-\Delta}\right)$.

This guarantees that the computed values are correct. To ensure that this computation requires only polynomial time, we need to compile ψ_Δ into an appropriate language, one that simultaneously supports TC, FO, and PR. According to Table 2, there is no such language, but the following theorem allows us to replace FO, not supported by cdn-MDAG, by $\mathtt{FO}_\mathtt{d}$, supported by cdn-MDAG.

Theorem 5. $\Delta \,||\, \psi_\Delta$.

As a consequence of this simple theorem, we arrive at the main message of this paper, namely that cdn-MDAG is the most suitable target compilation language for BNs, since it supports TC, $\mathtt{FO}_\mathtt{d}$, and PR, and thus allows to compute posterior probabilities in polynomial time.

For the compilation of the MCNF ψ_Δ into a cdn-MDAG, we can use the state-of-the-art CNF to d-DNNF or any CNF to OBDD complier [6,22] in the binary case. For the general case, we are currently working on the adaption of these algorithms.

4 Conclusion

The approach proposed in this paper extends a logical inference method for BNs with binary variables to BNs with multi-state variables. We expect its contribution to be theoretically and practically significant. On the theoretical side, based on an extended knowledge compilation map, the paper provides a precise explanation of why cdn-MDAGs are apparently the most suitable logical representations for BNs. This is mainly a consequence of the fact that some of the involved variables are deterministic. The paper also demonstrates how to reduce the problem of logical inference in BNs to three basic logical operations. Compared to Darwiche's differential approach, this view fits much better into the picture of the knowledge compilation map, as the reduction to these essential elements no longer requires us to talk about network polynomials, multi-linear functions, partial derivatives, arithmetic circuits, or smoothness. In this sense, we also see our paper as an attempt to clarify the theoretical mechanisms and connections behind this kind of inference algorithms and as a good example to demonstrate the usefulness of the knowledge compilation map.

On the practical side, the paper provides precise step-by-step instructions to implement a new encoding and inference method for BNs in terms of a few simple operations for cdn-MDAGs. Compared to Darwiche's differential approach, this will lead to more transparent implementations. Finally, with respect to possible applications other than BNs, other situations with deterministic variables may be detected, for which forgetting becomes tractable in the case of cdn-MDAGs.

Acknowledgment. Research is supported by the *Swiss National Science Foundation*, Project No. PP002-102652/1, and *The Leverhulme Trust*. Special thanks to Adnan Darwiche and the anonymous reviewers for their very helpful comments.

References

1. Guo, H., Hsu, W.H.: A survey of algorithms for real-time Bayesian network inference. In: Joint Workshop on Real-Time Decision Support and Diagnosis Systems (2002)
2. Sang, T., Beame, P., Kautz, H.: Solving Bayesian networks by weighted model counting. In: 20th National Conference on Artificial Intelligence, vol. 1 (2005)
3. Darwiche, A.: A differential approach to inference in Bayesian networks. Journal of the ACM 50(3), 280–305 (2003)
4. Chavira, M., Darwiche, A.: Compiling Bayesian networks using variable elimination. In: 20th International Joint Conference on Artificial Intelligence (2007)
5. Darwiche, A.: A logical approach to factoring belief networks. In: KR'02, 8th International Conference on Principles and Knowledge Representation and Reasoning, Toulouse, France, pp. 409–420 (2002)
6. Darwiche, A.: A compiler for deterministic, decomposable negation normal form. In: 18th National Conference on Artificial Intelligence, pp. 627–634 (2002)
7. Boutilier, C., Friedman, N., Goldszmidt, M., Koller, D.: Context-specific independence in Bayesian networks. In: 12th Conference on Uncertainty in Artificial Inteligence, pp. 115–123 (1996)
8. Chavira, M., Darwiche, A.: Compiling Bayesian networks with local structure. In: 19th International Joint Conference on Artificial Intelligence (2005)
9. Chavira, M., Darwiche, A., Jaeger, M.: Compiling relational Bayesian networks for exact inference. International Journal of Approximate Reasoning 42(1-2) (2006)
10. Wachter, M., Haenni, R.: Logical compilation of Bayesian networks. Technical Report iam-06-006, University of Bern, Switzerland (2006)
11. Wachter, M., Haenni, R.: Multi-state directed acyclic graphs. In: 20th Canadian Conference on Artificial Intelligence. LNAI, vol. 4509, pp. 464–475 (2007)
12. Darwiche, A., Marquis, P.: A knowledge compilation map. Journal of Artificial Intelligence Research 17, 229–264 (2002)
13. Wachter, M., Haenni, R.: Propositional DAGs: a new graph-based language for representing Boolean functions. In: KR'06, 10th International Conference on Principles of Knowledge Representation and Reasoning, pp. 277–285 (2006)
14. Palacios, H., Bonet, B., Darwiche, A., Geffner, H.: Pruning conformant plans by counting models on compiled d-DNNF representations. In: 15th International Conference on Planning and Scheduling, pp. 141–150 (2005)
15. Boole, G.: The Laws of Thought. Walton and Maberley (1854)
16. Kohlas, J.: Information Algebras: Generic Stuctures for Inference. Springer, Heidelberg (2003)
17. Davis, S., Putnam, M.: A computing procedure for quantification theory. Journal of the ACM 7(3), 201–215 (1960)
18. Darwiche, A.: Decomposable negation normal form. Journal of the ACM 48(4), 608–647 (2001)
19. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Mateo (1988)
20. Haenni, R.: Towards a unifying theory of logical and probabilistic reasoning. In: 4th International Symposium on Imprecise Probabilities and Their Applications, pp. 193–202 (2005)
21. Haenni, R., Kohlas, J., Lehmann, N.: Probabilistic argumentation systems. In: Handbook of Defeasible Reasoning and Uncertainty Management Systems, Algorithms for Uncertainty and Defeasible Reasoning, vol. 5, pp. 221–288 (2000)
22. Darwiche, A.: New advances in compiling CNF to decomposable negational normal form. In: 16th European Conference on Artificial Intelligence (2004)

Local Monotonicity in Probabilistic Networks

Johan Kwisthout*, Hans Bodlaender, and Gerard Tel

Department of Information and Computer Sciences, University of Utrecht,
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
{johank, hansb, gerard}@cs.uu.nl

Abstract. It is often desirable that a probabilistic network is monotone, e.g., more severe symptoms increase the likeliness of a more serious disease. Unfortunately, determining whether a network is monotone is highly intractable. Often, approximation algorithms are employed that work on a local scale. For these algorithms, the monotonicity of the arcs (rather than the network as a whole) is determined. However, in many situations monotonicity depends on the ordering of the values of the nodes, which is sometimes rather arbitrary. Thus, it is desirable to order the values of these variables such that as many arcs as possible are monotone. We introduce the concept of local monotonicity, discuss the computational complexity of finding an optimal ordering of the values of the nodes in a network, and sketch a branch-and-bound exact algorithm to find such an optimal solution.

1 Introduction

In many probabilistic networks [9] that are used for classification in real problem domains, the variables of the network can be distinguished into observable input variables, non-observable intermediate variables and a single output variable. For example, in a medical domain the observable variables represent clinical evidence such as observable symptoms and test results, the output variable functions as a classification of a disease, and the intermediate variables model non-observable facts that are relevant for classification. Often, the relations between observable symptoms and the classification variable are monotone, e.g., higher values for the observable variable 'fever' makes higher values of the classification variable 'flu' more likely, independent of the value of other variables such as 'headache'. Such a network is *monotone in distribution* [10] if higher-ordered configurations of the observable variables make higher-ordered outputs more (isotone) or less (antitone) likely.

When a domain expert indicates that a certain relation ought to be monotone, the joint probability distribution should be such, that this property is reflected in the network. If monotonicity is violated, the probability distribution in the network can be revised in cooperation with the expert. Unfortunately, determining whether a network is monotone in distribution is, in general, highly

* The work of this author was partially supported by the Netherlands Organisation for Scientific Research NWO.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 548–559, 2007.

intractable ([10]). One approach to overcome this unfavorable complexity, is by approximating the decision (i.e, sometimes have 'undecidable' as outcome) like the algorithm discussed in [10]. This algorithm uses qualitative influences (see e.g. [12] for an introduction in qualitative networks or QPNs) that summarize the direction of the influence of variables by signs. However, the use of these signs of course requires an ordering on the values of the variables under consideration. Such an ordering might be implicit, for example *large* > *medium* > *small* or *true* > *false*. But in practice, there are often variables in a network which do not have such 'natural' orderings. As it is desirable to have as many as possible monotone influences (to minimize the offending context), it is important to *choose* an ordering for the values of these variables that maximizes the number of monotone arcs. Or, equivalently, minimizes the number of '?' signs in the corresponding QPN.

In this paper, we determine the computational complexity of this optimization problem. In Section 2, we introduce some notations and definitions. We show that optimizing the number of monotone arcs is NP-complete and hard to approximate (Section 3). We suggest a branch-and-bound strategy as an exact algorithm in Section 4. Finally, we conclude our paper in Section 5.

2 Preliminaries

Let $\mathbf{B} = (G, \Gamma)$ be a Bayesian network where $G = (V, A)$ is an acyclic directed graph, and Γ, the set of conditional probability distributions, is composed of rational probabilities. Let Pr be the joint probability distribution of $\mathbf{B}$. The conditional probability distributions in Γ are assumed to be explicit, i.e., represented with look-up tables. For any variable $X \in V(G)$, let $\Omega(X)$ denote the set of values that X can take. A node X is denoted as a predecessor of Y if $(X, Y) \in A(G)$. The set of all predecessors of Y is denoted as $\pi(Y)$. If, for a node Y, $\pi(Y)$ is the set $X = \{X_1, \ldots, X_n\}$, the *configuration template* $\mathbf{X}$ is defined as $\Omega(X_1) \times \ldots \times \Omega(X_n)$; a particular instantiation $\mathbf{x}$ of $X_1, \ldots, X_n$ will be denoted as a *configuration* of $\mathbf{X}$.

2.1 Local Monotonicity

Monotonicity can be defined as stochastical dominance (monotone in distribution) or in a modal sense (monotone in mode). In this paper, we discuss monotonicity in distribution only, and we focus on local effects, i.e., influences between two variables which are directly connected. A network is *locally monotone* if all qualitative influences along the arcs in the network are either positive or negative.

Definition 1 (local monotonicity). *Let F be the cumulative distribution function for a node $X \in V(G)$, defined by $F(x) = Pr(X \leq x)$ for all $x \in \Omega(X)$. For any arc $(X, Y) \in A(G)$, let $\mathbf{Z}$ denote the configuration template $\pi(Y) \setminus X$, and let $\mathbf{z}$ denote an individual configuration of $\mathbf{Z}$. With (X, Y), a positive influence is associated if $x < x' \rightarrow F(y \mid x\mathbf{z}) \geq F(y \mid x'\mathbf{z})$ for all $y \in \Omega(Y), x, x' \in \Omega(X)$, and $\mathbf{z} \in \mathbf{Z}$. Similarly, a negative influence is associated with this arc if $x < x' \rightarrow$*

$F(y \mid x\mathbf{z}) \leq F(y \mid x'\mathbf{z})$ *for all* $y \in \Omega(Y), x, x' \in \Omega(X)$, *and* $\mathbf{z} \in \mathbf{Z}$. *We will denote an arc associated with an positive or negative influence as a isotone, respectively antitone arc.* $\boldsymbol{B} = (G, \Gamma)$ *is locally monotone if all arcs in* $A(G)$ *are either isotone or antitone.*

2.2 Interpretations

The above notions of monotonicity assumed an implicit *ordering* on the values of the variables involved. Such an ordering is often trivial (e.g., $x > \bar{x}$ and *always* > *sometimes* > *never*) but sometimes it is arbitrary, like an ordering of the values { *trachea*, *mediastinum*, *diaphragm*, *heart* }. Nevertheless, a certain ordering is necessary to determine whether the network is monotone, or to determine which parts of the network are violating monotonicity assumptions. Thus, for nodes where no *a priori* ordering is given, we want to order the values of these nodes in a way that maximizes the number of monotone arcs or the number of nodes with only monotone incoming arcs (depending on the specific application).

We define the notion of an *interpretation* of X to denote a certain ordering on $\Omega(X)$, the set of values of X. Note, that the number of distinct interpretations of a node with k values equals $k!$, the number of permutations of these values. Nevertheless, in practice, the number of values a variable can take is often small. For example, in the ALARM network [2], the number of values is at most four, and in the OESOPHAGEAL network [11] it is at most six. In this paper, we assume that k is small and can be regarded as a fixed constant.

Definition 2 (interpretation). *An interpretation of* $X \in V(G)$, *denoted* I_X, *is a total ordering on* $\Omega(X)$. *For arbitrary interpretations we will often use* σ *and* τ. *We use the superscript* T *to denote a reverse ordering: if* $\sigma = (x_1 < x_2 < \ldots < x_n)$, *then* $\sigma^T = (x_n < \ldots < x_2 < x_1)$. *The interpretation set* $\mathbf{I_X}$ *is defined as the set of all possible interpretations of* X. *Note that an arc is isotone for a given interpretation* σ *if and only if it is antitone for* σ^T *and vice versa, and that the interpretations in* $\mathbf{I_X}$ *are pairwise symmetric. In the remainder, when* $\sigma, \tau \in \mathbf{I_X}$ *are distinct, then we also assume that* $\sigma \neq \tau^T$.

2.3 Monotonicity Functions and Schemes

We define a *monotonicity function*, which determines whether a certain combination of interpretations for the two nodes of an arc makes the arc isotone or antitone. When a node has more than one predecessor (say $\pi(Y) = \{X_1, X_2\}$), the arc (X_1, Y) is monotone for a certain combination of interpretations $\sigma \in \mathbf{I}_{X_1}$ and $\tau \in \mathbf{I}_Y$, when it is isotone for all values[1] of X_2, or when it is antitone for all values of X_2. We define the monotonicity function of (X_1, Y) for a particular *given* value $x_2 \in \Omega(X_2)$ as a *partial monotonicity function*, to emphasise the *conditional* monotonicity of (X_1, Y).

[1] Note that the *ordering* of the elements in $\Omega(X_2)$ is irrelevant for the local monotonicity of (X_1, Y).

Definition 3 ((partial) monotonicity function). *Consider the arc* $x_1 = (X_1, Y) \in A(G)$*, where* Y *has auxiliary predecessors (say* $x_2 \ldots x_n$*), whose configuration template we denote with* $\mathbf{Z}_N$*. Assume* $\sigma \in \mathbf{I}_{X_1}$ *and* $\tau \in \mathbf{I}_Y$*. Then* $M_{X_1Y}(\sigma, \tau)$ *is true if and only if* x_1 *is either isotone (denoted* $M^+_{X_1Y}$*) or antitone (denoted* $M^-_{X_1Y}$*) for interpretations* σ *and* τ*, for all possible configurations of* $\mathbf{Z}_N$*. The partial monotonicity function* $M_{X_1Y}(\sigma, \tau \,|\, \mathbf{z}_N)$ *is true if and only if* x_1 *is isotone or antitone for interpretations* σ *and* τ*, given a specific configuration* $\mathbf{z}_N$ *of* $\mathbf{Z}_N$*.*

Observe, that $M_{XY}(\sigma, \tau) = M_{XY}(\sigma^T, \tau) = M_{XY}(\sigma, \tau^T) = M_{XY}(\sigma^T, \tau^T)$ since $M_{XY} = M^+_{XY} \vee M^-_{XY}$, and $M^+_{XY}(\sigma, \tau) \leftrightarrow M^-_{XY}(\sigma^T, \tau)$. Partial monotonicity functions and schemes can be combined for multiple configurations of $\mathbf{Z}_N$. Informally, the combined partial monotonicity function for instantiation x_ϕ and x_ψ is true for a certain combination of interpretations, if the individual partial monotonicity functions are all isotone, or all antitone, for that combination.

Definition 4 (combining partial monotonicity functions). *Consider again the arc* x_1 *as defined before, with* $\mathbf{Z}_N$ *as the configuration template of* $\pi(Y) \setminus X_1$*. Then, for* $\delta \in \{+, -\}$*,*

$$M^\delta_{X_1Y}(\sigma, \tau \,|\, \mathbf{z}_\phi) \wedge M^\delta_{X_1Y}(\sigma, \tau \,|\, \mathbf{z}_\psi) = M^\delta_{X_1Y}(\sigma, \tau \,|\, \mathbf{z}_\phi \wedge \mathbf{z}_\psi)$$

and consequently,

$$\bigwedge_{z_N \in Z_N} M^\delta_{X_1Y}(\sigma, \tau \,|\, \mathbf{z}_N) = M^\delta_{x_1Y}(\sigma, \tau)$$

With every monotonicity function M_{XY}, a binary matrix $\mathbf{M_{XY}}$ is associated, denoted as the *monotonicity scheme* of M_{XY}. Similarly, a *partial monotonicity scheme* $\mathbf{M_{XY|z_N}}$ is associated with the corresponding partial monotonicity function. These matrices have dimensions $\frac{1}{2} \mid \mathbf{I}_X \mid \times \frac{1}{2} \mid \mathbf{I}_Y \mid$, since the interpretations in $\mathbf{I}$ are pairwise symmetric. We will often illustrate these matrices using a grid, where shaded areas denote monotone combinations of interpretations in $\mathbf{I}_X$ and $\mathbf{I}_Y$.

Using these definitions, in the following section we will discuss the problem of optimizing the number of monotone arcs, i.e., choosing an interpretation for the values of all nodes, such that the number of arcs that are isotone or antitone is maximal. Note that a network where some nodes are fixed (i.e., an interpretation is given) can be translated in an equivalent network with non-fixed interpretations, where the number of monotone arcs will be optimal if and only if that particular interpretation is chosen. For example, if the ordering of a node C with values $\{low, mid, high\}$ and degree n is to be fixed at $low < mid < high$, we can enforce this condition by adding $n + 1$ dummy nodes D with $\Omega(D) = \{T, F\}$ and arcs from C to these nodes that are only monotone if C has ordering $low < mid < high$, for example $\Pr(T \,|\, low) = 0.2$, $\Pr(T \,|\, mid) = 0.4$, $\Pr(T \,|\, high) = 0.6$. It can be easily verified that the optimal number of monotone arcs enforces the given ordering on C. In a similar way,

a partial order can be guaranteed, e.g., the variable 'Stereo Sound' with values $\{none, left, right, both\}$ where no obvious ordering for '*left*' and '*right*' exists, but $none \prec left \prec both$ and $none \prec right \prec both$.

3 Optimizing the Number of Monotone Arcs

In this section, we formalize the problem of optimizing the number of monotone arcs, and show that it is NP-complete, i.e., infeasible in general. A similar complexity result is established for the derived problem of optimizing the number of nodes with only monotone incoming arcs. Both problems can be used as a measure for the size of the monotonicity-violating context. Furthermore, we prove that these problems — apart from infeasible to solve exactly — are hard to approximate as well. In the remainder of this section, we assume that the reader is familiar with NP-completeness proofs; more background can be found in textbooks like [6] and [7].

In the formal problem definitions, we assume that the (conditional) probabilities in the network are specified using rationals, rather than reals, to ensure an efficient coding of these probabilities. Since these probabilities are often specified by experts or approximated using learning methods, this is a realistic constraint. Furthermore, we assume that the conditional probabilities in the network are coded explicitly, i.e., using look-up tables, rather than using some computable function. Lastly, for technical reasons we formulate our problems as decision problems (returning 'yes' or 'no'), rather than functions (returning a number).

MAX-LOCAL MONOTONICITY

Instance: Let $\mathbf{B} = (G, \Gamma)$ be a Bayesian network where Γ is composed of rational probabilities, and let Pr be its joint probability distribution. Let $\Omega(X)$ denote the set of values that $X \in V(G)$ can take, and let k be a positive integer $\leq | A(G) |$.

Question: Is there an interpretation I_X for all $X \in V(G)$, such that the number of arcs in G that are monotone in distribution is at least k?

MAX-NODES-LOCAL MONOTONICITY

Instance: Let $\mathbf{B} = (G, \Gamma)$ and $\Omega(X)$ be as above, and let k be a positive integer $\leq | V(G) |$.

Question: Is there an interpretation I_X for all $X \in V(G)$, such that the number of nodes in G that have only incoming arcs that are monotone in distribution, is at least k?

In our hardness proof, we use the GRAPH 3-COLORABILITY problem, defined in [6]. In this problem, the instance is an undirected graph $G = (V, E)$, and we want to know whether there is a function $f : V \to \{1, 2, 3\}$, such that $f(u) \neq f(v)$ whenever $(u, v) \in E$, i.e., all nodes can be colored with three colors, such that no adjacent nodes have the same color.

3.1 NP-Completeness Proof

Let $G = (V, E)$ be an instance of the Graph 3-Colorability problem. From this undirected graph G, we construct the directed graph $G' = (V', A)$ as follows (See Figure 1):

- if $X \in V(G)$, then $X \in V'$.
- if $(X, Y) \in E(G)$, then $E_1, E_2, E_3, E_4, E_5, E_6 \in V'$.
- if $(X, Y) \in E(G)$, then $(X, E_1), \ldots, (X, E_6), (Y, E_1), \ldots, (Y, E_6) \in A$.
- $\Omega(X) = \{x_1, x_2, x_3\}$ for all $X \in V'$

We number the interpretations of all nodes in V' as follows:

- $i_1 = x_2 < x_1 < x_3$.
- $i_2 = x_1 < x_2 < x_3$.
- $i_3 = x_1 < x_3 < x_2$.

Now, for all nodes E_i we construct a conditional probability table such that $M(I_X, I_{E_i})$ has the following monotonicity scheme:

E_i	E_1	E_2	E_3	E_4	E_5	E_6
I_X	$\{i_1, i_2\}$	$\{i_1, i_2\}$	$\{i_1, i_3\}$	$\{i_1, i_3\}$	$\{i_2, i_3\}$	$\{i_2, i_3\}$
I_{E_i}	$\{i_2, i_3\}$	$\{i_1, i_3\}$	$\{i_1, i_2\}$	$\{i_2, i_3\}$	$\{i_1, i_2\}$	$\{i_1, i_3\}$

and the probability table for the arc (Y, E_i) is such, that (Y, E_i) is a monotone relation if and only if $I_Y = I_{E_i}$. An example of such a table is given in Table 1; the other tables can be generated likewise. Observe, that a graphical representation of these schemes would be a 2×2 square, which is *transposed* from the origin. We claim that, in the thus constructed network, there is a maximum of eight arcs that have a monotone relation, if $I_X = I_Y$, and nine arcs if $I_X \neq I_Y$. We assume, without loss of generality, that $I_Y = i_1$. If we choose $I_{E_i} = e_1$ for all E_i, then all six outgoing arcs from Y to E_i have monotone relations. Now there are two cases:

- $I_X = i_1$. There are two monotone relations: (X, E_2) and (X, E_3). Both E_2 and E_3 have only monotone incoming arcs.
- $I_X = i_2$ or i_3. There are *three* monotone relations: either (X, E_2), (X, E_5) and (X, E_6); or (X, E_3), (X, E_5) and (X, E_6), which all have only monotone incoming arcs.

Note that there is no way to make *more* than three monotone arcs. We will use this construct to prove NP-hardness.

Theorem 1. Max-Local Monotonicity *and* Max-Nodes-Local Monotonicity *are NP-complete.*

Proof. Membership of NP is trivial for both problems. Using a certificate that consists of interpretations for all vertices, we can easily test whether at least k arcs are monotone in distribution, or at least k nodes have the property that all

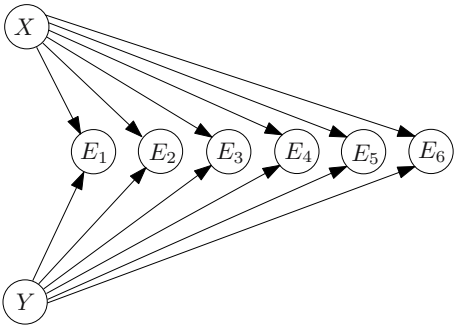


Fig. 1. Construction with 6 extra nodes

Table 1. Conditional probability table for node E_1 with incoming arcs from X and Y

$\mathbf{Pr}(E \mid y_1, X)$			
	e_1	e_2	e_3
x_1	0.42	0.30	0.28
x_2	0.28	0.44	0.28
x_3	0.28	0.30	0.42

$\mathbf{Pr}(E \mid y_2, X)$			
	e_1	e_2	e_3
v_1	0.44	0.28	0.28
v_2	0.28	0.44	0.28
v_3	0.28	0.28	0.44

$\mathbf{Pr}(X \mid y_3, X)$			
	e_1	e_2	e_3
v_1	0.44	0.28	0.28
v_2	0.28	0.44	0.28
v_3	0.28	0.28	0.44

incoming arcs are monotone in distribution. To prove *NP*-hardness, we construct a transformation from the Graph 3-Colorability problem. Let $G = (V, E)$ be an instance of this problem, and let $G' = (V', A)$ be the directed acyclic graph the is constructed from this instance, as described above. If and only if $9\times \mid E \mid$ arcs in G' are monotone, then all nodes X and Y that were adjacent in G, have different interpretations, hence G would be 3-colorable. Since $G' = (V', A)$ can be computed from $G = (V, E)$ in polynomial time, we have a polynomial-time transformation from Graph 3-Colorability to the Max-Local Monotonicity problem, which proves NP-hardness of the latter. A similar argument holds for the number of nodes with only monotone incoming arcs, therefore Max-Nodes-Local Monotonicity is NP-hard as well. □

3.2 Approximation

When confronted with intractable (e.g., NP-hard) problems, a number of options are available. One could try to find polynomial algorithms for particular instances (special cases) of the problem, try to construct an exact algorithm that works reasonably fast most of the time, or try to approximate the problem. In the latter case, one tries to find a solution that is *close* to optimal, in reasonable time. Not all problems are easy to approximate. An example of an NP-hard problem which can be approximated in polynomial time within an arbitrary margin is Scheduling Independent Tasks [3]: given a set of tasks of variable length and a set of processors on which they run, which schedule leads to a minimum total finishing time? Other problems are very hard to approximate. For example,

the well known Travelling Salesman Problem cannot be approximated in general within any fixed constant factor, unless P = NP.

Optimization problems can be classified based on the character of the *performance ratio* of their approximation algorithms. We will give a short introduction on this classification; for a more thorough introduction the reader can refer to e.g. [8], [5], or [1]. For maximization problems, the ratio $R(X, Y)$ of an approximation algorithm Y, given instance X, is defined as $R(X, Y) = \frac{OPT(X)}{APP_Y(X)}$, where $OPT(X)$ denotes the optimal solution for X, and $APP_Y(X)$ denotes the solution, given by algorithm Y. The class of all optimization problems is denoted with NPO. A subset of this class is the class APX. A problem A belongs to APX if it is approximable within a fixed ratio, i.e. there is an algorithm T and a ratio r, such that for all instances X, $R(X, T) \leq r$. A problem A belongs to PTAS (has a polynomial time approximation scheme) if it is approximable within any ratio r in time polynomial in the input size, and it belongs to FPTAS (has a *fully* polynomial time approximation scheme) if this approximation is polynomial in r as well.

In this section, we will show that Max-Local Monotonicity is APX-hard. This hardness result is a very strong indicator that there is no polynomial time approximation scheme for it – otherwise, *all* problems in APX would enjoy such a PTAS – and that the problem can only be approximated within a constant factor. We will reduce Max-3color-Subset [8], a known APX-hard problem, to Max-Local Monotonicity using a so-called *A-reduction*. In [4], an A-reduction is defined as a reduction from A to B, such that an approximation for B within a fixed ratio r implies an approximation for A within a fixed ratio $c(r)$, where c is a computable function $\mathbf{Q} \cap (1, \infty) \rightarrow \mathbf{Q} \cap (1, \infty)$. If there is an A-reduction from A to a known APX-hard problem B, then A is APX-hard as well. We will show that the reduction from 3color constructed in the previous section is actually an A-reduction to Max-3color-Subset.

Theorem 2. Max-Local Monotonicity *A-reduces to* Max-3color-Subset.

Proof. Let $G = (V, E)$ be an instance of Max-3color-Subset, and let $G' = (V', A)$ be the directed acyclic graph the is constructed from this instance, as described in Section 3.1. Let $OPT(3C)$ denote the maximum number of nodes in G that can be colored with three colors, and let $APP_Y(3C)$ be the number of nodes colorable with three colors with a certain approximation algorithm Y. The ratio r of this approximation is then

$$\frac{OPT(3C)}{APP_Y(3C)}$$

By construction, G' has an optimal solution $8 \mid V \mid + OPT(3C)$, and Y would approximate this to $8 \mid V \mid + APP_Y(3C)$, with ratio

$$r' = \frac{8 \mid V \mid + OPT(3C)}{8 \mid V \mid + APP_Y(3C)}$$

But then, there clearly exists a function c such that $r \leq \phi \Rightarrow r' \leq c(\phi)$. □

Corollary 1. MAX-LOCAL MONOTONICITY *is APX-hard.*

4 A Branch-and-Bound Algorithm

In the previous section we proved that there does not exist a PTAS for MAX-LOCAL MONOTONICITY unless APX = PTAS. However, there might exist approximations for MAX-LOCAL MONOTONICITY that are within a fixed ratio r. Nevertheless, r may be very large and such approximations may not be particularly useful. Therefore, we now construct an exact algorithm for this problem, based on a so-called branch-and-bound strategy (see for example [13]). In such a strategy, the set of possible solutions is partitioned (the branch step), and upper (or lower, for minimalization problems) bounds for this partition are calculated. Whenever these bounds are lower than or equal to the current best solution (i.e., further exploration of these branches will not lead to a better solution) the branch is terminated, and other, yet unvisited branches are explored. This procedure continues until all branches terminate (we can return an optimal solution), or a given ratio between current best solution and upper bound is reached (we can return a 'good enough' solution).

4.1 Initial Heuristic - A Lower Bound

In this section we discuss how a lower bound on the number of monotone arcs can be calculated in polynomial time (for fixed k). First we will present a procedure to compute monotonicity schemes efficiently, exploiting a particular property of monotonicity functions. We will distinguish between *factorizing* and *non-factorizing* monotonicity schemes, and we will show how a lower bound heuristic can be calculated, using arcs with factoring monotonicity schemes, in polynomial time.

Trivially, computing a monotonicity scheme for any node takes $O((k!)^2)$, since there are $k!$ interpretations for both ends of the arc, and computing local monotonicity takes linear time. Notice that the complexity of calculating schemes for an arc whose endpoint has multiple predecessors, is proportional in the size of the input (i.e., the conditional probability table). If all other variables in the graph have an arc towards this endpoint, there are $\prod_{i=1}^{n} \mid X_i \mid$ configurations that we need to consider. However, the conditional probability table has size $O(\prod_{i=1}^{n} \mid X_i \mid)$ as well, since we assumed explicit probability representation.

This running time can be reduced to $O(\frac{1}{2}(k!)^2)$ in the worst case, and $O(2(k!))$ in the best case, by exploiting the following observation. If a relation $X \rightarrow Y$ is monotone for two distinct interpretations $\sigma, \sigma' \in \mathbf{I}_X$ given an interpretation $\tau \in \mathbf{I}_Y$, then there are at least two equal columns in the joint probability table, i.e., $\Pr(y_k \,|\, x_i) = \Pr(y_k \,|\, x_j)$ for all $y_k \in \Omega(Y)$. But then, $M_{XY}(\sigma, \tau') = M_{XY}(\sigma', \tau')$ for *all* interpretations $\tau' \in \mathbf{I}_Y$. Of course, in two distinct interpretations[2] there exist i and j such that $x_i \leq x_j$ in one interpretation and $x_j \leq x_i$ in the other. From this property follows, that two columns in a monotonicity scheme are

[2] Note that we defined σ and σ' to be distinct, only if also $\sigma' \neq \sigma^T$.

either equal or disjoint. It suffices to observe that there exists a $\tau \in \mathbf{I}_Y$ such that $M_{XY}(\sigma, \tau) = M_{XY}(\sigma', \tau) = \text{TRUE}$ to conclude that this is the case for *all* $\tau \in \mathbf{I}_Y$.

To compute a lower bound heuristic, we consider only arcs that have *factorizing* monotonicity schemes. We use these factorizing schemes to calculate *allowed sets* for each variable.

Definition 5 (factorizing monotonicity scheme). $\mathbf{M_{XY}}$ *is called factorizing over* $\mathbf{I}_X$ *and* $\mathbf{I}_Y$ *if there exist subsets* $\mathbf{I}^+_\mathbf{X} \subseteq \mathbf{I_X}$ *and* $\mathbf{I}^+_\mathbf{Y} \subseteq \mathbf{I_Y}$ *such that* $M_{XY}(\sigma, \tau)$ *is true if and only if* $\sigma \in \mathbf{I}^+_\mathbf{X}$ *and* $\tau \in \mathbf{I}^+_\mathbf{Y}$.

For a variable Z, let $\pi(Z)$ denote its set of predecessors and let $\sigma(Z)$ denote its set of children. Then, if all arcs $(X \in \pi(Z), Z)$ and $(Z, Y \in \sigma(Z))$ have a factoring monotonicity scheme, an interpretation I_Z for Z that is an element of $\bigcap_{X \in \pi(Z)} \mathbf{M_{XZ}} \cap \bigcap_{Y \in \sigma(Z)} \mathbf{M_{ZY}} \neq \emptyset$ is always an interpretation that can be chosen for Z without violating local monotonicity of the network. Of course, not all monotonicity schemes are factoring. If $\pi(Z)_f$ and $\sigma(Z)_f$ denote the predecessors, respectively children of Z such that $(X \in \pi(Z)_f, Z)$, respectively $(Z, Y \in \sigma(Z)_f)$ are arcs with factoring monotonicity schemes, we will denote $\mathcal{M}_Z$ as the *allowed set* of Z, where $\mathcal{M}_Z = \bigcap_{X \in \pi(Z)_f} \mathbf{M_{XZ}} \cap \bigcap_{Y \in \sigma(Z)_f} \mathbf{M_{ZY}} \cap \mathbf{I_Z}$. Note, that the allowed set consists of interpretations that can be chosen, if all arcs *without* factoring monotonicity schemes would be removed. In other words, there exists a network $G' = (V, A')$ where A' is the (possibly empty) set of arcs with factoring monotonicity schemes, and the allowed set of all $Z \in V(G')$ is the set of interpretations that can be chosen without violating monotonicity of G'. Now, we can calculate a lower bound for the maximal number of arcs in G that can be made monotone as follows. We initialise $\mathcal{M}_Z$ to $\mathbf{I_Z}$ for all $Z \in V'$ and A^+ to the empty set, and iteratively consider arcs in A'. If an arc does not cause any allowed set to become empty, it is added to A^+, and $\mathcal{M}$ is adapted for both endpoints of that arc. On the other hand, if the arc does lead to an empty allowed set, it is dismissed. After considering all arcs in A', $\mid A^+ \mid$ is a lower bound.

4.2 Branching and Bounding

Using this lower bound, we consecutively branch on the possible interpretations of the nodes, terminating branches whose upper bound is not higher than the current best solution (or lower bound). While different strategies can be followed to choose a node to branch on at any step in the algorithm, a reasonable heuristic is to pick the node that has the highest degree of all unexplored nodes. We fix the interpretation of the variable we branch on (i.e., the allowed set is a singleton, corresponding with the branch value) and calculate how many factorizing arcs remain monotone in the network. This value is added to the number of non-factorizing arcs; this is an upper bound for the total number of monotone arcs in the network.

Of course, there are many degrees of freedom in this branch-and-bound strategy. We chose to compute rather loose bounds; one can compute tighter bounds

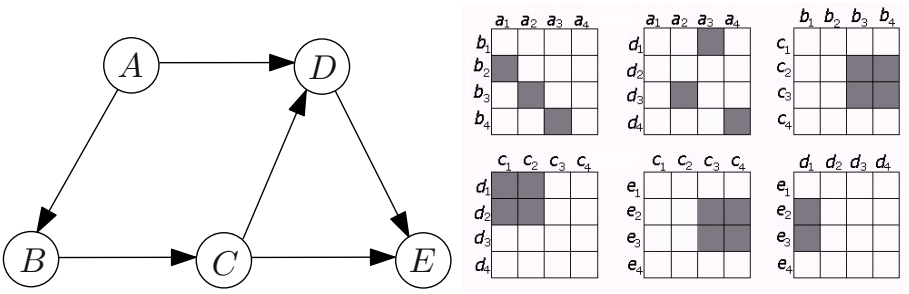


Fig. 2. An example graph

by considering a number of non-factoring arcs that can be made monotone. Nevertheless, the constraints imposed by these arcs might require re-evaluation of all allowed sets in the network, so there is a tradeoff between the tightness of the bounds - and thus the number and depth of the branches - and the time needed to calculate such bounds.

4.3 An Example

We will use the graph in Figure 2 as a example to sketch our branch-and-bound algorithm. We assume that, for every variable, only four interpretations are relevant; we will denote a particular interpretation with indexed lowercase variables, e.g., $\mathbf{I_C} = \{c_1, c_2, c_3, c_4\}$. On the right part of Figure 2, the monotonicity schemes for the arcs in the graph are shown. For example, (A, B) is monotone if $I_A = a_1$ and $I_B = b_2$.

We start with the heuristic lower bound calculated in Section 4.1. The factorizing arcs are (B, C), (C, D), (C, E), and (D, E), and if we consider these in this order and calculate the allowed sets for all nodes, we will find that we can make at least three arcs monotone, namely (B, C), (C, D), and (D, E). The lower bound will thus be three in this example. Now we branch on one of the nodes with maximal degree, say C, and explore the branches $I_C = c_1$, $I_C = c_2$, $I_C = c_3$, and $I_C = c_4$, terminating branches with an upper bound lower than three. Eventually, the algorithm will find the optimal solutions $\{I_A = a_3, I_B = b_4, I_C = c_3, I_D = d_1, I_E = e_2 \vee e_3\}$.

5 Conclusion

Optimising the number of monotone arcs in a network, and thus minimising the number of '?'s in the corresponding QPN, is a computationally hard problem, and hard to approximate as well. We proposed a branch-and-bound approach to calculate optimal orderings. This approach may work rather well in practice with 'real world' networks, provided that the number of values per node is small. However, for networks where some nodes have a large range of possible values,

this approach will be infeasible. Other methods must be used in such cases to calculate or approximate an optimal solution. Currently, we are working on an implementation of this branch-and-bound algorithm.

Acknowledgements

This research has been (partly) supported by the Netherlands Organisation for Scientific Research (NWO). The authors wish to thank Linda van der Gaag and Silja Renooij for their insightful comments on this subject and access to the information stored in the Oesophageal Cancer network, and Jesper Nederlof for his work on the implementation of the algorithm. Furthermore, we wish to thank three anonymous reviewers for their useful comments on an earlier version of this paper.

References

1. Ausiello, G., Crescenzi, P., Protasi, M.: Approximate solution of NP optimization problems. Theoretical Compututer Science 150(1), 1–55 (1995)
2. Beinlich, I., Suermondt, G., Chavez, R., Cooper, G.: The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In: Proceedings of the Second European Conference on AI and Medicine, Springer, Heidelberg (1989)
3. Bruno, J., Coffman jr, E.G., Sethi, R.: Scheduling independent tasks to reduce mean finishing time. Communications of the ACM 17(7), 382–387 (1974)
4. Crescenzi, P.: A short guide to approximation preserving reductions. In: 12th Annual IEEE Conference on Computational Complexity (CCC'97), pp. 262–273. IEEE, Los Alamitos (1997)
5. Crescenzi, P., Panconesi, A.: Completeness in approximation classes. Information and Computation 93, 241–262 (1991)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability. In: A Guide to the Theory of NP-Completeness, W. H. Freeman and Co, San Francisco (1979)
7. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, Reading (1994)
8. Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation, and complexity classes. Journal of Computer and System Sciences 43, 425–440 (1991)
9. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, Palo Alto (1988)
10. van der Gaag, L.C., Bodlaender, H.L., Feelders, A.: Monotonicity in Bayesian networks. In: Twentieth Conference on Uncertainty in Artificial Intelligence, pp. 569–576. AUAI Press (2004)
11. van der Gaag, L.C., Renooij, S., Witteman, C.L.M., Aleman, B.M.P., Taa, B.G.: Probabilities for a probabilistic network: a case study in oesophageal cancer. Artificial Intelligence in Medicine 25, 123–148 (2002)
12. Wellman, M.P.: Fundamental concepts of qualitative probabilistic networks. Artificial Intelligence 44(3), 257–303 (1990)
13. Wolsey, L.A., Nemhauser, G.L.: Integer and Combinatorial Optimization. In: Discrete Mathematics and Optimization. Wiley-Interscience Series, Wiley, Chichester (1988)

Independence Decomposition in Dynamic Bayesian Networks

Ildikó Flesch and Peter Lucas

Department of Information and Knowledge Systems
Institute for Computing and Information Science
Radboud University Nijmegen, The Netherlands
{ildiko,peterl}@cs.ru.nl

Abstract. Dynamic Bayesian networks are a special type of Bayesian network that explicitly incorporate the dimension of time. They can be distinguished into repetitive and non-repetitive networks. Repetitiveness implies that the set of random variables of the network and their independence relations are the same at each time step. Due to their structural symmetry, repetitive networks are easier to use and are, therefore, often taken as the standard. However, repetitiveness is a very strong assumption, which normally does not hold, as particular dependences and independences may only hold at certain time steps.

In this paper, we propose a new framework for independence modularisation in dynamic Bayesian networks. Our theory provides a method for separating atemporal and temporal independence relations, and offers a practical approach to building dynamic Bayesian networks that are possibly non-repetitive. A composition operator for temporal and atemporal independence relations is proposed and its properties are studied. Experimental results obtained by learning dynamic Bayesian networks from real data show that this framework offers a more accurate way for knowledge representation in dynamic Bayesian networks.

1 Introduction

Probabilistic graphical models are increasingly adopted as tools for the modelling of domains involving uncertainty. For the development of practical applications especially Bayesian networks have gained much popularity. When considering these application domains, it appears that so far only limited attention has been given to the modelling of uncertain time-related phenomena, which occur in many of these domains. Bayesian networks in which some notion of time is explicitly dealt with are usually called *dynamic* Bayesian networks (DBNs) [3]. In some domains involving time, such as speech recognition, the use of special DBNs has been extensively explored (e.g. [1]), and technical issues such as concerning reasoning (e.g. [7]) and learning (e.g. [2]) in DBNs have been investigated.

DBNs are distinguished into two main classes: repetitive and non-repetitive networks. Repetitive networks have the same set of random variables and independence relations at each time step, whereas in non-repetitive networks the set

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 560–571, 2007.

of random variables and also the independence relations between these random variables may vary in time. The simpler structure of repetitive networks provides significant advantages in terms of ease of modelling and computational complexity. Therefore, they are often seen as the standard DBN model (see [6] for an overview). However, repetitiveness is a very strong assumption that normally will not hold.

Recently, scientific evidence has become available that non-repetitive DBNs may also be practically useful [8]. We think that separating temporal and atemporal information in DBNs can be valuable, as it: (*i*) helps experts gain more insight into the relations in the networks, (*ii*) provides an opportunity for learning procedures to obtain more accurate models, and (*iii*) may help overcome computational limitations. However, so far no research has been carried out to characterise temporal and atemporal independence relations.

In this paper, a new framework for independence modularisation in DNBs is proposed, based on a theoretically grounded separation of temporal and atemporal independences. This distinction allows us to investigate isolated parts of the independence relations. Having given these individual parts of the network, we can construct both repetitive and non-repetitive DBNs. In this paper, we analyse the necessary properties to correctly join independence relations from these individual parts and define a join operator to carry out the composition of these independence relations. Finally, we provide experimental evidence of the usefullness of non-repetitive DBNs.

2 Motivating Example: The Disease Course of VAP

A real-world non-repetitive DBN of the disease course of a form of pneumonia is used as motivating example in this paper. As we will see, at each time step we have different independence relations offering an accurate model of the evolution of the disease.

We briefly describe the clinical features of pneumonia and then discuss the construction of a DBN for this disease. Pneumonia develops frequently in ICU patients, as these patients are critically ill and often they need respiratory support by a mechanical ventilator. After admission to a hospital, all patients become colonised by bacteria. In particular, mechanically ventilated patients run the risk of subsequently developing pneumonia caused by these bacteria; this type of pneumonia is known as *ventilator-associated pneumonia*, or VAP for short. Typical signs and symptoms of VAP include: high body temperature, decreased lung function (measured by the PaO_2/FiO_2 ratio) and evidence of pneumonia on the chest X-ray. By carrying out a dependency analysis on a retrospective, temporal dataset, with data of ICU patients collected during a period of three years, we were able to study how independence information changed in the course of time. Taking the duration of mechanical ventilation as the parameter defining the time steps, we have focused on modelling the course of the development of VAP at 3, 4 and 5 days after admission. The resulting DBN is shown in Fig. 1.

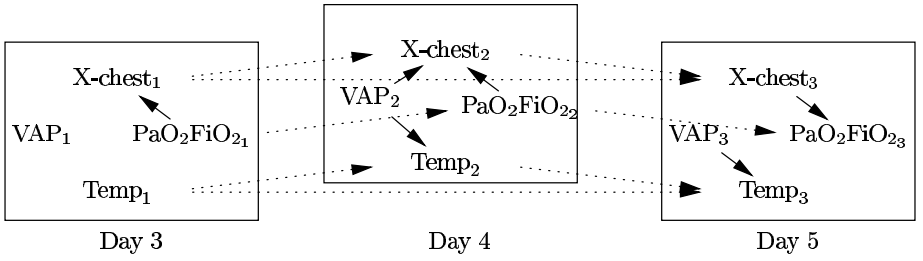


Fig. 1. The non-repetitive DBN for VAP; temporal arcs are depicted by dotted arrows

This small network already shows the need for employing non-repetitiveness in the structure of a DBN, and, therefore, has motivated us to develop a theory that distinguishes between temporal and atemporal independence information and allows building a non-repetitive DBN in a correct and seamless fashion.

3 Basic Notions

We will be concerned in this paper with acyclic directed graphs (ADGs), denoted as a pair $G = (V, A)$, where V is a set of vertices and $A \subseteq V \times V$ is a set of *arcs*. A *directed path* is a sequence of vertices $v_1, v_2, \ldots, v_m$, with $(v_k, v_{k+1}) \in A$ for each k, also denoted by $v_k \rightarrow v_{k+1}$, where $v_1, v_2, \ldots, v_{m-1}$ are required to be distinct. A *directed cycle* is a directed path with $v_1 = v_m$. A *trail* θ in a graph is a sequence of unique vertices $v_1, v_2, \ldots, v_m$, where we have for each k that $v_k \rightarrow v_{k+1}$ or $v_{k+1} \rightarrow v_k$; each arc occurs only once. A *subtrail* of a trail $v_1, v_2, \ldots, v_m$ is a sequence $v_i, v_{i+1}, \ldots, v_j$, $i < j$. A trail θ connecting vertices u and v is also written as $u \sim v$. The set of all trails of an ADG G is denoted by Θ. A graph $G'_{|\Theta'} = (V', A')$ is said to be a *reduced subgraph* of graph $G = (V, A)$ with associated set of trails Θ if $V' \subseteq V$, A' consists of all arcs of the set of trails Θ' with $\Theta' \subseteq \Theta$, and Θ' is based on the set of vertices V'.

Let X be a set of *discrete* random variables and let V act as its index set, i.e., X_v with $v \in V$ denotes a random variable and X_W with $W \subseteq V$ denotes a set of random variables. Furthermore, let P denote a joint probability distribution (JPD) of X_V. The set X_U is said to be *conditionally independent* of X_W *given* X_Z, with $U, W, Z \subseteq V$, if

$$P(X_U \mid X_W, X_Z) = P(X_U \mid X_Z) \; . \tag{1}$$

These independence relations in P can also be represented by means of an ADG G, in which the entire set of independence relations is denoted by $\perp\!\!\!\perp_G$. In the graph, arcs represent dependences, and absence of arcs represents (conditional) independences.

Independences can be read off from an ADG by the *d-separation criterion*, defined as follows [5]: a trail θ in an ADG G is said to be *blocked* by a set Z if

one of the following conditions is satisfied: (*i*) $v \in Z$ and v appears on the trail θ, and either *no* or *only one* of the arcs of θ meeting at v is directed to v; (*ii*) $v \notin Z$, $\delta(v) \cap Z = \varnothing$, where $\delta(v)$ are the descendants of v, and *both* arcs meeting at v on θ are directed to v (*convergent connection*). It is said that the sets U and W are *d-separated* by Z if any trail between a vertex in U and a vertex in W is blocked by the set Z; formally: $U \perp\!\!\!\perp_G W \mid Z$. Otherwise, U and W are *d-connected* by Z, denoted by $U \not\perp\!\!\!\perp_G W \mid Z$.

A *Bayesian network* is defined as a pair $\mathcal{B} = (G, P)$, where $G = (V, A)$ is an acyclic directed graph representing relations of random variables X_V, P is the JPD on X_V, and each independence represented in G is also a valid independence in the JPD P.

4 Dynamic Bayesian Networks

In this section, the foundation for independence modularisation in DBNs is developed, based on the separation of temporal and atemporal independences.

DBNs are an extension of ordinary Bayesian networks and allow modelling the uncertainty involved in time-oriented processes. As a start, a representation of *time* is required, which in this paper is denoted by T and is assumed to be a subset of the set of the natural numbers; a *time point* t is then a member of T. The graphical representation of a DBN consists of two parts: (*i*) an atemporal part, and (*ii*) a temporal part. We subsequently define these parts.

An acyclic directed graph $G_t = (V_t, A_t^a)$, with set of vertices V_t and set of arcs $A_t^a \subseteq V_t \times V_t$, $t \in T$, is called a *timeslice* at time t, and its set of arcs A_t^a is called the set of *atemporal arcs*. Timeslices will be depicted by rectangles. The set of all timeslices G of a DBN is taken as:

$$G = \{G_t \mid t \in T\} = (V_T, A^a) \; . \tag{2}$$

Let $G_t = (V_t, A_t^a)$ and $G_{t'} = (V_{t'}, A_{t'}^a)$, $t, t' \in T$, $t \neq t'$, be two distinct timeslices. Then, an arc $(u, v) \in V_t \times V_{t'}$ with $t < t'$ is called a *temporal arc*. The set of temporal arcs of the set of all time slices G is denoted by A^t. Thus, temporal arcs connect vertices in different timeslices; they direct *always* from the past to the future and are drawn as dotted arrows.

Example 1. Consider Fig. 2; here the set of timeslices is equal to $G = \{G_1, G_2, G_3\}$. Timeslice G_2 is defined as $G_2 = (V_2, A_2^a)$, where the set of its vertices is equal to $V_2 = \{q_2, s_2, v_2, z_2\}$ and the set of its atemporal arcs $A_2^a = \{(q_2, s_2), (v_2, z_2)\}$). Moreover, the temporal arcs are equal to $A^t = \{(u_1, v_2), (w_1, v_2), (v_2, r_3)\}$.

Temporal arcs connect timeslices; this allows to construct temporal networks.

Definition 1. ***(temporal network)*** *Let $G = (V_T, A^a)$ be a set of timeslices. Then, a* temporal network *N is defined as a pair $N = (V_T, A)$, where $A = A^a \cup A^t$.*

Clearly, as each timeslice $G_t \in G$ is an acyclic directed graph, and timeslices are connected by temporal arcs pointing from the past to the future, N is also an

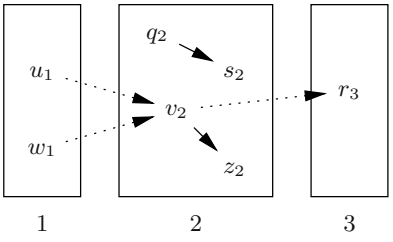


Fig. 2. An example of a dynamic Bayesian network

acyclic directed graph. A DBN is defined as a pair $\mathcal{DBN} = (N, P)$, where P is the JPD on the entire set of random variables.

Both temporal and atemporal relations in the network can be represented by means of trails. An *atemporal trail* contains no temporal arcs and is denoted by θ^a. A *temporal trail* consists of *at least one* temporal arc and is denoted by θ^t. The sets of all atemporal and temporal trails are denoted by Θ^a and Θ^t, respectively.

Example 2. Fig. 2 includes temporal trails $\theta_1^t = u_1 \rightarrow v_2 \leftarrow w_1$ and $\theta_2^t = z_2 \leftarrow v_2 \rightarrow r_3$, and the only two atemporal trails are $\theta_1^a = q_2 \rightarrow s_2$ and $\theta_2^a = v_2 \rightarrow z_2$.

With regards to the temporal relationships we only need to consider temporal trails that result into a reduced temporal network.

Definition 2. ***(reduced temporal network)*** *Let $N = (V_T, A)$ be a temporal network. Then, $N_{|\Theta^t} = (V_T, A_{\Theta^t})$ is called a* reduced temporal network *if its set of arcs $A_{\Theta^t} \subseteq A$ consists of all the arcs included on the temporal trails in Θ^t.*

Observe that the reduced temporal network is based on the set of temporal *trails*, which may consist of both atemporal and temporal trails. A further partitioning of the reduced temporal network is based on its set of *arcs*. This partitioning is obtained by decomposing a reduced temporal network into two parts, where one part consists of only atemporal and the another part of only temporal arcs. The *atemporal part of the reduced temporal network* is denoted by $N^a_{|\Theta^t} = (V_T, A^a_{\Theta^t})$, where V_T is the set of vertices and $A^a_{\Theta^t} \subseteq A^a$ consists of all atemporal arcs in the reduced temporal network. The *temporal part of the reduced temporal network* is denoted by $N^t_{|\Theta^t} = (V_T, A^t_{\Theta^t})$, where $A^t_{\Theta^t} \subseteq A^t$ consists of all temporal arcs in the reduced temporal network.

As a DBN includes temporal and atemporal elements, the question is how to distinguish between these relations. Vertices U and W are said to be *atemporally d-separated* (*temporally d-separated*) by Z, denoted by $\perp\!\!\!\perp_G$ ($\perp\!\!\!\perp_{N_{|\Theta^t}}$), if all the atemporal (temporal) trails connecting U and W are d-separated given Z. Atemporal d-separation of vertices belonging to only one timeslice G_t is denoted by $\perp\!\!\!\perp_{G_t}$. Finally, $\perp\!\!\!\perp_N$ denotes the set of independences in the temporal network N.

Fig. 3 shows an example of a temporal network and includes the various temporal and atemporal parts defined above.

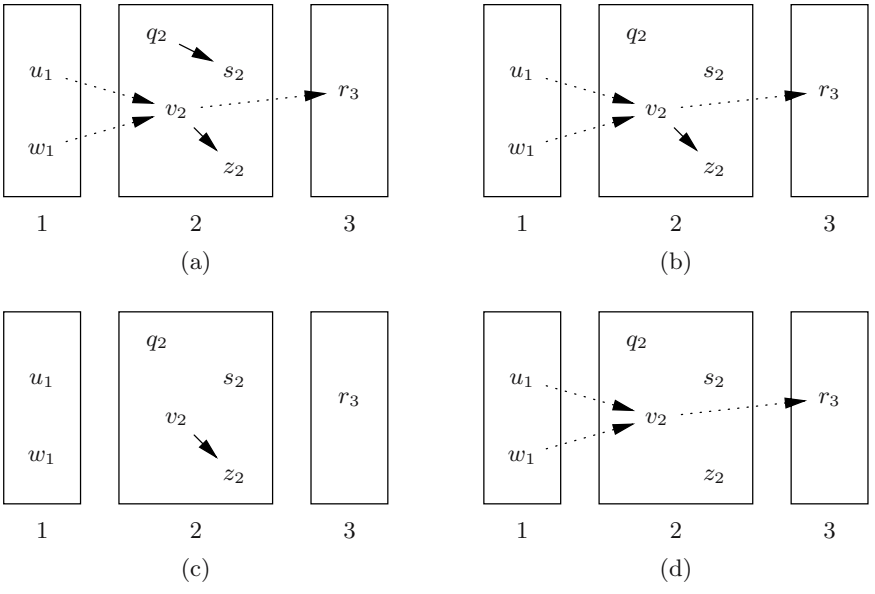


Fig. 3. Temporal and atemporal parts of a temporal network: (a) temporal network, (b) reduced temporal network, (c) atemporal part of the reduced temporal network, and (d) temporal part of the reduced temporal network

5 The Join Operator

In this section, we study the correct composition of the temporal and atemporal independence relations in DBNs.

Suppose we want to join two independence relations $\perp\!\!\!\perp$ and $\perp\!\!\!\perp'$, both defined on the same set of variables. Then, the following three situations need to be considered: (*i*) joining one dependence and one independence statement; (*ii*) joining two dependence statements; (*iii*) joining two independence statements.

To correctly join the statements from situations (*i*) and (*ii*) a dependence preservation property is proposed below. Similarly, in order to deal with situation (*iii*), an independence concatenation property is defined below.

5.1 Dependence Preservation

The reason that dependence preservation is required for joining dependence and independence or dependence and dependence statements can be explained in terms of the concepts of consistency and dominance; these terms are interpreted as follows.

Let the independence relations $\perp\!\!\!\perp$ and $\perp\!\!\!\perp'$ be defined on the same vertex set V. Then, if there exist statements $U \perp\!\!\!\perp W \mid Z$ and $U \not\perp\!\!\!\perp' W \mid Z$ for arbitrary, mutually disjoint sets of vertices $U, W, Z \subseteq V$, then these independence statements and, therefore, independence relations $\perp\!\!\!\perp$ and $\perp\!\!\!\perp'$ are said to be *inconsistent*.

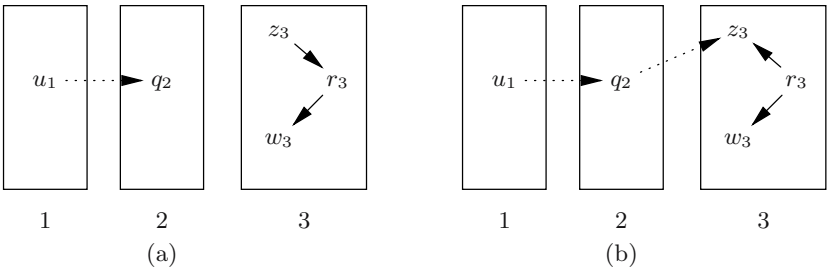


Fig. 4. Temporal networks (a) and (b)

Otherwise, the statements are *consistent*. If we wish to join independence relations together, and two independence relations are inconsistent, a choice has to be made between the independence and dependence. In other words, one statement has to *dominate* the other one. If the relations $\perp\!\!\!\perp$ and $\perp\!\!\!\perp'$ are inconsistent due to the statements $U \perp\!\!\!\perp W \mid Z$ and $U \not\perp\!\!\!\perp' W \mid Z$, then $U \not\perp\!\!\!\perp' W \mid Z$ is said to *dominate* $U \perp\!\!\!\perp W \mid Z$. Furthermore, the dominance of dependences also indicates that two dependence statements, mentioned for situation (*ii*), also have to be joined into a dependence statement. Since dominance has to be taken into account when joining independence relations, the following property is defined.

Definition 3. ***(dependence preservation)*** *Let $\perp\!\!\!\perp$, $\perp\!\!\!\perp'$ and $\perp\!\!\!\perp''$ be independence relations all defined on V. Suppose that $U \not\perp\!\!\!\perp W \mid Z$ or $U \not\perp\!\!\!\perp' W \mid Z$, or both, then it holds that $U \not\perp\!\!\!\perp'' W \mid Z$ for all $U, W, Z \subseteq V$. It is said that $\perp\!\!\!\perp''$ satisfies the* dependence preservation *property with regard to $\perp\!\!\!\perp$ and $\perp\!\!\!\perp'$.*

5.2 Independence Concatenation

The independence concatenation property takes into account how independence relations are combined.

Definition 4. ***(independence concatenation)*** *Let $\perp\!\!\!\perp$, $\perp\!\!\!\perp'$ and $\perp\!\!\!\perp''$ be independence relations all defined on V. Suppose that for some $U, W, Z \subseteq V$ independences $U \perp\!\!\!\perp W \mid Z$ and $U \perp\!\!\!\perp' W \mid Z$ hold and then it also holds that $U \perp\!\!\!\perp'' W \mid Z$. Then, it is said that $\perp\!\!\!\perp''$ satisfies the* independence concatenation *property with regard to the set of vertices U, W and Z in relations $\perp\!\!\!\perp$ and $\perp\!\!\!\perp'$.*

In the remaining part of this section, we investigate how the independence concatenation property can be applied to ADGs. The following example indicates that joining independences is *not* straightforward in these graphs, since an independence may change into a dependence.

Example 3. Consider Fig. 4. In both the atemporal and temporal parts of (a) and (b) vertex u_1 is conditionally independent of w_3 given z_3; formally, we have that $u_1 \perp\!\!\!\perp_{N^a_{|\Theta^t}} w_3 \mid z_3$ and $u_1 \perp\!\!\!\perp_{N^t_{|\Theta^t}} w_3 \mid z_3$. In the reduced temporal network of (a) u_1 is still conditionally independent of w_3 given z_3; however, in the reduced

temporal network of (b), u_1 is conditionally dependent of w_3 given z_3; formally: $u_1 \perp\!\!\!\perp_{N_{|\Theta^t}} w_3 \mid z_3$ and $u_1 \not\perp\!\!\!\perp_{N_{|\Theta^t}} w_3 \mid z_3$.

Thus, it is necessary to investigate how to join two independence statements. As mentioned above, in Bayesian networks d-separation criterion is used to read-off independence statements, based on the study of the arc directions in the trails in the graphical representations. Therefore, when we want to join two independence relations $\perp\!\!\!\perp_G$ and $\perp\!\!\!\perp_{G'}$ into $\perp\!\!\!\perp_{G''}$, we need to investigate the trails in graph G'' obtained from the set of trails of the graphs G and G'. Then, any trail in G'' can be partitioned into a set of subtrails, called *special subtrails*, where each subtrail consists of arcs obtained from only *one* of the graphs G or G'.

Example 4. Reconsider Fig. 3 and suppose we want to join the atemporal and temporal parts (c) and (d) of the reduced temporal network obtaining the entire independence relation of the reduced temporal network (b). Suppose we want to determine the relation between vertices u_1 and z_2. Then, we need to study trail $u_1 \rightarrow v_2 \rightarrow z_2$ in graph (b), which consists of two special subtrails, atemporal trail $v_2 \rightarrow z_2$ in (c) and temporal trail $u_1 \rightarrow v_2$ in (d).

Considering the issues mentioned above, when we need to join two independence statements, we only need to analyse trails in G'' that consist of more than one special subtrail; otherwise the relation is just an independence statement, since its arcs belong only to one of the graphs G or G'.

After having singled out the set of special subtrails of a trail, we can determine the independence relations from the trail by looking at these subtrails. This is done by examining the independence and dependence statements, derived from $\perp\!\!\!\perp_G$ and $\perp\!\!\!\perp_{G'}$, between the initial and end vertex of each special subtrail, while conditioning on the same set of vertices as from the entire trail. Then, if at least one of these two statements is a dependence, according to the dependence preservation property, the initial and end vertex of the special subtrail will be conditionally dependent in G''. However, in the case of two independence statements we need to apply d-separation. Furthermore, we also need to consider the set of so-called *shared vertices* of a trail, which are the vertices that connect the special subtrails of this trail. These considerations give rise to an easier way of the composition of two independence statements:

Proposition 1. *Let the independence relations $\perp\!\!\!\perp_G$, $\perp\!\!\!\perp_{G'}$ and $\perp\!\!\!\perp_{G''}$ both defined on V and let $U \perp\!\!\!\perp_G W \mid Z$, $U \perp\!\!\!\perp_{G'} W \mid Z$ for $U, W, Z \subseteq V$ hold in graphs $G = (V, A)$ and $G' = (V, A')$, respectively. Let $A'' \subseteq A \cup A'$. Let trail θ connect the two vertices $u \in U$ and $w \in W$ with each other, and let $\theta_1, \ldots, \theta_n$ be the special subtrails of θ. Then, θ is* blocked *by Z if one of the following condition holds:*

- *the trail θ consists of only one special subtrail;*
- *for one of the special subtrails $\theta_i = v_1, v_2, \ldots, v_m$ we have $v_1 \perp\!\!\!\perp_G v_m \mid Z$ and $v_1 \perp\!\!\!\perp_{G'} v_m \mid Z$ and θ_i is blocked by Z in G'' according to d-separation;*
- *one of the shared vertices blocks θ according to d-separation.*

The independence $U \perp\!\!\!\perp_{G''} W \mid Z$ holds if each trail connecting any vertex in U and W is blocked by Z satisfying the independence concatenation property.

5.3 The Join Operator

Next, the join operator is defined and a significant property of this operator is considered.

Definition 5. *(**join operator**) Let* $\perp\!\!\!\perp$ *and* $\perp\!\!\!\perp'$ *be two independence relations defined on the same vertex set* V*. The* join *of these two relations, denoted by* $\perp\!\!\!\perp \circ \perp\!\!\!\perp' = \perp\!\!\!\perp''$*, is then again an independence relation,* $\perp\!\!\!\perp''$*, defined on* V*, that satisfies the dependence preservation and the independence concatenation properties.*

Proposition 2. *Let* $G = (V, A), G' = (V, A')$ *and* $G'' = (V, A'')$ *be three ADGs, where* $A \cup A' \subseteq A''$*. Then, it holds that* $\perp\!\!\!\perp_{G''} \subseteq \perp\!\!\!\perp_G \circ \perp\!\!\!\perp_{G'}$*.*

Proof. The graph G'' contains at least as many arcs as the union of the graphs G and G'. Thus, $\perp\!\!\!\perp_{G''}$ contains possibly extra dependences in addition to those in $\perp\!\!\!\perp_G$ and $\perp\!\!\!\perp_{G'}$. As the join operator satisfies the independence concatenation property, any independence that results from applying d-separation to G'' is preserved by joining $\perp\!\!\!\perp_G$ and $\perp\!\!\!\perp_{G'}$. Hence, it follows that $\perp\!\!\!\perp_{G''} \subseteq \perp\!\!\!\perp_G \circ \perp\!\!\!\perp_{G'}$. □

6 Temporal and Atemporal Interaction

Based on the results above, in this section we investigate how to employ the join operator for combining the temporal and atemporal relations underlying temporal networks to support the modelling of *non-repetitive* DBNs.

6.1 Joining Atemporal and Reduced Temporal Networks

In this section, we start by considering the relations $\perp\!\!\!\perp_G$ and $\perp\!\!\!\perp_{G_t}$. The following proposition establishes that the join operator $\circ$ can be interpreted as the intersection of the independence relations in G_t.

Proposition 3. *Let* $\mathcal{DBN} = (N, P)$ *with temporal network* $N = (V_T, A)$*, set of timeslices* $G = (V_T, A^a)$*,* $A = A^a \cup A^t$*, and the joint probability distribution* P*. Then, it holds that:*

(i) $\perp\!\!\!\perp_G = \cap_{t \in T} \perp\!\!\!\perp_{G_t}$*;*
(ii) $\not\perp\!\!\!\perp_G = \cup_{t \in T} \not\perp\!\!\!\perp_{G_t}$*.*

The graph-theoretic interpretation of dependence preservation and independence concatenation for reduced temporal networks is investigated next.

Proposition 4. *Let* $N^a_{|\Theta^t}$ *and* $N^t_{|\Theta^t}$ *be the atemporal and temporal parts of reduced temporal network* $N_{|\Theta^t}$*. Then, the dependence preservation and independence concatenation properties hold for the independence relations* $\perp\!\!\!\perp_{N^a_{|\Theta^t}}$*,* $\perp\!\!\!\perp_{N^t_{|\Theta^t}}$ *and* $\perp\!\!\!\perp_{N_{|\Theta^t}}$*.*

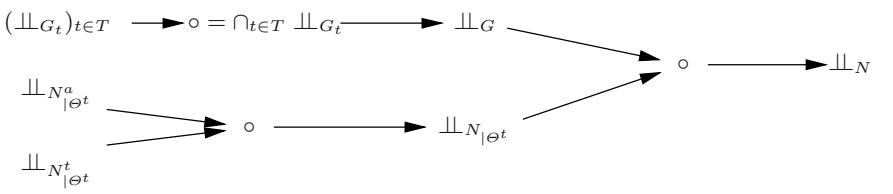


Fig. 5. Joining temporal and atemporal independence relations

Next we will show that the join operator can be used to merge the two independence relations, proving its soundness and completeness. *Soundness* of the join operator means that all independence statements obtained by joining two independence relations can be read off from the union of the underlying graphs, whereas *completeness* means that none of the independence statements of the union of the graphs has been omitted in the resulting independence relation.

Theorem 1. *Let* $\perp\!\!\!\perp_{N^a_{|\Theta^t}}$, $\perp\!\!\!\perp_{N^t_{|\Theta^t}}$, *and* $\perp\!\!\!\perp_{N_{|\Theta^t}}$ *be constructed as defined above. Then, it holds that* $\perp\!\!\!\perp_{N_{|\Theta^t}} = \perp\!\!\!\perp_{N^a_{|\Theta^t}} \circ \perp\!\!\!\perp_{N^t_{|\Theta^t}}$, *i.e. the join operator is sound and complete.*

Proof. *Soundness* follows from the independence concatenation property. By this property only independence statements that hold can be derived, because it is based on Proposition 1 and, therefore, on temporal d-separation. *Completeness* follows from Proposition 2 by substituting $N^a_{|\Theta^t}$, $N^t_{|\Theta^t}$, and $N_{|\Theta^t}$ for acyclic directed graphs G, G' and G'', respectively. □

6.2 Joining It All Together

In this subsection, the temporal and atemporal independence relations are joined together, yielding the relation $\perp\!\!\!\perp_N$. To start, the following proposition and theorem show that these relations can be linked to each other by means of the join operator.

Proposition 5. *Let* G *and* $N_{|\Theta^t}$ *be the atemporal and temporal parts of temporal network* N. *Then, the dependence preservation and independence concatenation properties hold for the independence relations* $\perp\!\!\!\perp_G$, $\perp\!\!\!\perp_{N_{|\Theta^t}}$ *and* $\perp\!\!\!\perp_N$.

Theorem 2. *Let* $\perp\!\!\!\perp_G$, $\perp\!\!\!\perp_{N_{|\Theta^t}}$ *and* $\perp\!\!\!\perp_N$ *be constructed as defined above. Then, it holds that* $\perp\!\!\!\perp_N = \perp\!\!\!\perp_G \circ \perp\!\!\!\perp_{N_{|\Theta^t}}$, *i.e. the join operator is sound and complete.*

Proof. *Soundness* and *completeness* are similar to those of Theorem 1. □

Fig. 5 provides a summary of propositions 3, 4 and 5. Finally, the various independence relations can be compared to each other.

Proposition 6. *The following properties hold:*

- $\perp\!\!\!\perp_{N_{|\Theta^t}} \subseteq \perp\!\!\!\perp_{N^a_{|\Theta^t}}$ *and* $\perp\!\!\!\perp_{N_{|\Theta^t}} \subseteq \perp\!\!\!\perp_{N^t_{|\Theta^t}}$;
- $\perp\!\!\!\perp_N \subseteq \perp\!\!\!\perp_G$ *and* $\perp\!\!\!\perp_N \subseteq \perp\!\!\!\perp_{N_{|\Theta^t}}$.

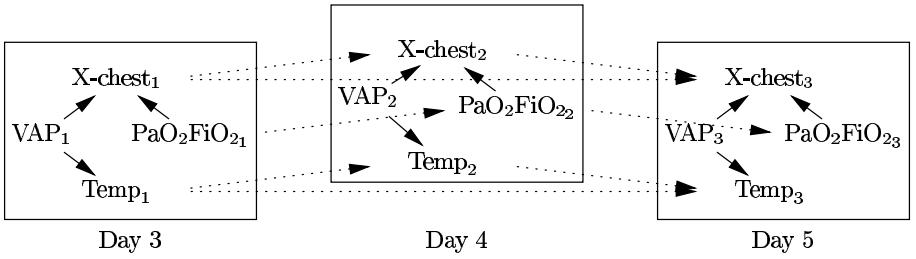


Fig. 6. The repetitive DBN for VAP

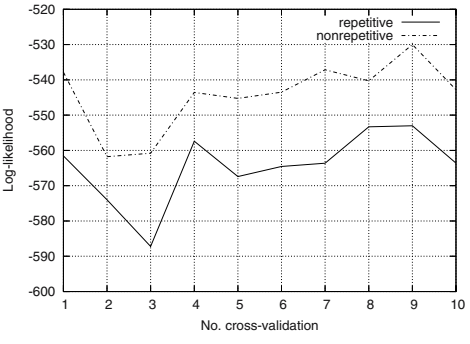


Fig. 7. The log-likelihoods of the repetitive and non-repetitive DBNs

7 Experimental Results

Using available temporal data of 876 patients with and without VAP, we experimentally compared repetitive and non-repetive DBNs with the aim of demonstrating the usefullness of the framework. The experiments were done using Murphy's BNT toolbox [6].

First, based on the non-repetitive DBN of progression of VAP, shown in Fig. 1, a repetitive DBN was constructed; it is shown in Fig. 6. The graphical structure of the non-repetitive network was obtained by using the join operator described in Section 6. Evaluation of the quality of the two DBNs was done using stratified tenfold cross-validation, i.e., the VAP dataset was divided randomly into 10 parts, where 9 parts acted as training sets, whereas the remaining part acted as a test set. The entire cross-validation process was repeated 10 times. In the process, the probability distributions of the DBNs were estimated using the EM algorithm [4].

As a measure of the quality of the two DBNs use was made of the log-likelihood function l [4]:

$$l(\mathcal{B}; D) = \sum_{d \in D} \log P_{\mathcal{B}}(d),$$

where $\mathcal{B}$ denotes a DBN, D denotes the test set of data of patients from the ICU with and without VAP, and $d \in D$ denotes a tuple with patient data. The higher the value of the log-likelihood is, the better the distribution fits to the data. For each iteration of cross-validation, the average log-likelihood was computed.

The experimental results obtained are shown in Fig. 7. On the abcis are the iteration numbers of individual tenfold cross-validation processes. Although the log-likelihood varied for each of the DBNs between the different iterations, the log-likelihood for the non-repetitive DBN is *always* better than for the repetitive DBN. Thus, there exist datasets for which non-repetitive DBNs outperform repetitive DBNs, at the same time offering a more natural and precise form of knowledge representation.

8 Conclusions

The aim of the research described in this paper was to develop a framework for modelling non-repetitive and repetitive DBNs, based on a modularisation of independence relations. It appeared that by distinguishing between temporal and atemporal independence relations, modelling DBNs can be greatly facilitated.

How to build a DBN from its atemporal and temporal parts is not obvious. This problem was tackled by the introduction of a join operator with special semantics. Using the join operator allows one to build DBNs in a modular fashion, which in particular is important when designing non-repetitive DBNs. As far as we know, our paper offers the first systematic method for building non-repetitive DBNs.

The practical usefulness of the method in the context of learning DBNs from data was also explored, using a real-world problem, and evidence of the validity of the method was obtained.

References

1. Deviren, M., Daoudi, K.: Continuous speech recognition using dynamic Bayesian networks: a fast decoding algorithm. In: Proc PGM'02, Spain, pp. 54–60 (2002)
2. Friedman, N., Murphy, K., Russell, S.: Learning the structure of dynamic probabilistic networks. In: Proc 14th UAI, pp. 139–147 (1998)
3. Jensen, F.V.: Bayesian Networks and Decision Graphs. Springer, New York (2001)
4. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York (2001)
5. Cowell, R.G., Philip Dawid, A., Lauritzen, S.L., Spiegelhalter, D.J.: Probabilistic Networks and Expert Systems. Springer, New York (1999)
6. Murphy, K.P.: Dynamic Bayesian Networks: Representation, Inference and Learning. PhD Thesis, UC Berkeley (2002)
7. Kjaerulff, U.: A computational scheme for reasoning in dynamic probabilistic networks.In: Proc UAI'92, pp. 121–129 (1992)
8. Tucker, A., Liu, X.: Learning Dynamic Bayesian Networks from Multivariate Time Series with Changing Dependencies. IDA (2003)

Average and Majority Gates: Combining Information by Means of Bayesian Networks

L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, and M.A. Rueda-Morales

Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S.I. Informática, Universidad de Granada, 18071 – Granada, Spain
{lci,jmfluna,jhg,mrueda}@decsai.ugr.es

Abstract. In this paper we focus on the problem of belief aggregation, i.e. the task of forming a group consensus probability distribution by combining the beliefs of the individual members of the group. We propose the use of Bayesian Networks to model the interactions between the individuals of the group and introduce average and majority canonical models and their application to information aggregation. Due to efficiency restrictions imposed by the Group Recommending problem, where our research is framed, we have had to develop specific inference algorithms to compute group recommendations.

1 Introduction

In this paper we investigate the value of using Bayesian Networks (BN) to represent how different individuals in a group interact in order to achieve a final choice or recommendation. Although aggregating information is a common task to a number of disciplines, including statistics, decision theory, economics, political science, psychology, etc., our research is framed in the problem of *Group Recommending*, task included in the more general field of Recommending Systems (RS). The the objective is to obtain the most appropriate recommendations for groups of people where their members may be inter-related in different ways. In this problem it is usual to assume that the individuals do not have observed those items that might be recommended. This kind of RS is appropriate for domains where a group of people participates in a single activity such as watching a movie or going on holiday and also situations where a single person must make a decision about a group of people. This is a relatively novel problem (research started focusing on group recommending at the start of the 21st century [1,2]) and has hardly been researched in the literature.

Since the focus of this paper is on the *combination* of individuals opinions, we will discuss, non the process by which the individuals reach their opinion, neither the relationships between the members of the group. In this case we shall assume that all the individuals use the same set of labels to express their preferences on an item, and that these preferences are represented by means of a probability distribution (probably estimated from a data set). On the other hand, we will not discuss about subjects such as how the groups are formed or how long they have existed.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 572–584, 2007.

According to [3] *"... there is nothing close to a single well-accepted normative basis for group beliefs, group preferences or group decision making."* In this paper we show how different common decision rules in the literature could be managed by a proper design of canonical models with the language of BN, giving some new lights into the combination processes, particularly:

- The average strategy, *AVG*, which obtains the group rate as the average of the members' rates.
- The majority strategy, *MAJ*, which obtains the group rate as a simply counting of the votes that every group's member gives for an item. The final rate will be that one with more votes in the count.

The second section of this paper presents related work on information aggregation. Section 3 describes how to model the group interaction by means of BN topology, and presents the use of both average and majority gates. Section 4 presents some experimental results obtained when applying the proposed methodology to recommend movies for group of users. Finally, Section 5 includes our conclusions and some comments about further research.

2 Related Work

There are many papers focusing on combination of probabilistic information, ranging from a pure statistical approach (see [4,5] for a review) to more applied problems, as the combination of classifiers [6], prediction markets [7], various sources of information in a single BN [8] or different BNs into a unique model [3].

In general, the methods for combining information are dichotomized [4] into *mathematical* and *behavioural* approaches. Mathematical approaches consist of processes or analytical models that operate on the individual probability distributions to produce a single "combined" probability distribution; whereas behavioural approaches attempt to generate agreement among expert by having them interact in some way. Since this paper is focused on mathematical approaches we are going review those paper relevant to this subject.

Combining Probability Distributions: Mathematical approaches can be further distinguished into axiomatic approaches (considering a set of assumptions that the combination criteria might satisfy) and Bayesian approaches [4]:

- Axiomatic approach: Common functions to deal with belief aggregation are:
  i) *Linear Opinion Pool* where the probability of the group , $Pr(G)$, is obtained as the weighted arithmetic average over the probabilities of the individuals, $Pr(V_i), i = 1, \ldots, n$, i.e. $Pr(G) = \sum_{i=1}^{n} w_i Pr(V_i)$, w_i being weights summing one.
  ii) *Logarithmic Opinion Pool* (weighted geometric average) defined as $Pr(G) = \alpha \prod_{i=1}^{n} Pr(V_i)^{w_i}$, α being a normalization constant and the weights w_i (called expert weights) typically are restricted to sum one. If the weights are equal to $1/n$, then the combined distribution is proportional to geometric average.

– Bayesian Approach [5,4] has been used to combine expert information. This approach assumes that if there is a decision maker who has prior probability over the group vote Pr_0, and a likelihood function over individual opinions given the group vote, then, taking the individuals opinions as evidence the group priors over the pattern of vote can be updated according to Bayes rule. Usually, an in order to obtain efficient combinations, it is assumed that individuals opinion are conditionally independent given the group vote.

Group Recommending: Although the problem of group recommending is relatively new, the same dichotomy can be found, depending on whether they use individuals opinions to get to a consensus recommendations [9,10,11] or not [1,2,12]. In general, when focusing on "mathematical" approaches, ad hoc combinations criteria have been used. For instance [2], which selects the music stations to be played at a gym, computes the group preference for each (music) category by summing the squared individual preferences. Then, using a weighted random selection operator, the next music station to be played is selected.

Related to collaborative-based group RS is Polylens [1], which is an extension of the MovieLens [13] system that recommends movies to groups of users. These systems use nearest neighbour algorithms to find those individuals which are similar to group tastes and to obtain recommendations which merge the voting preferences of these individuals according to the principle of least misery (minimum criterion).

3 Modeling Group Decision Networks

As mentioned before, in this paper we shall not consider questions about how groups are formed nor how they are managed. We shall therefore assume that we know the composition of the groups, and our problem is to study how this information can be represented in the BN and also how to predict recommendations for groups, i.e. how the inference processes can be performed.

We propose to identify a group as an entity where recommendations are made by considering the particular recommendations of its members in some way. Figure 1 shows a typical situation where each member of the group has a guess about the probability of relevance of a given item. In order to model group behaviour we propose that the group node (G) has as parents ($Pa(G)$) the set of nodes representing its individuals (right hand side of Figure 1). In this paper we shall not discuss the case where the combination mechanism might be represented by means of a "naive-Bayes-like" approach, i.e. a root group node having as children the set of individuals. This modelization might be related to the classical Bayesian Approach for combining probability distributions. Following with our model, in Figure 1 we use dashed lines to represent the idea that the individuals opinion would be obtained by considering different pieces of information. It is interesting to note that we do not impose any restriction about the dependences or independences between the members of the group, i.e. the individuals might use information of some common variables when predicting

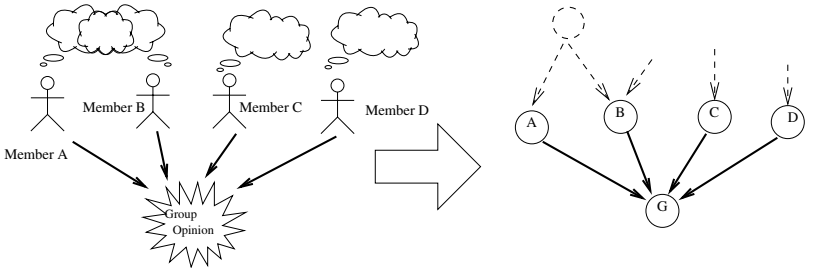


Fig. 1. Modeling group's interactions

their votes. We will only assume that the group opinion is independent of the information sources, $\mathcal{IS}$, given that we know the opinion of its members, i.e. $I(G|Pa(G)|\mathcal{IS})$.

With the idea of being general, the predicted rate for the i^{th} group, G_i, will be obtained by considering the individuals opinions, possibly updated after knowing new pieces of evidence, ev. Ideally, the opinion obtained by merging these individuals probabilities should represent the information of the group. Considering the proposed BN representation, the posterior probability of the group voting with rate s is obtained by means of

$$Pr(G_i = s|ev) = \sum_{pa(G_i)} Pr(G_i = s|pa(G_i)) \times Pr(pa(G_i)|ev).$$

being $pa(G_i)$ a particular configuration for the parent set of G_i, i.e an instance of its members. We shall denote by $\mathcal{R} = \{1, \ldots, r\}$ the set of possible rating alternatives. These conditional probability distributions can be considered as a "social value function" describing how the opinions of the members affect the group's recommendation.

One of our objectives is to obtain combining strategies able to be implemented in real-time situations. Therefore the reduction of the computational complexity becomes a key design parameter. Thus, since in real Group Recommending applications, the problem is to select those items, I_k, with higher probability of being liked by the group, i.e. for each pair *item–group* we have to compute the probability $Pr(G_i|I_k)$ and considering that there usually exists a large set of unobserved items in the system (which act as the evidence) this process becomes computationally expensive and therefore it is necessary to look for efficient algorithms. It is also interesting to note that with this requirement in mind, we propose to use canonical models to define the probabilities $P(G_i|pa(G_i))$. In general, since the group might be large, it implies important savings in storage (we do not need to store the exponential number of conditional probability values) and also efficient inference algorithms could be obtained.

3.1 Weighted Sum Gate

With this gate we are modeling that the group rate can be considered as the average vote of its members. This situation can be represented by means of the following additive canonical model, which is an extension of [14]:

$$P(G_i = k|pa(G_i)) = \sum_{j=1}^{|Pa(G_i)|} w(v_{j,s}, g_{i,k}), \tag{1}$$

where $w(v_{j,s}, g_{i,k})$ can be interpreted as the weight (effect) that the j^{th} group member voting the s^{th} value has in the k^{th} rate of the group G_i. The only restriction that we have to impose is that w are a set of non-negative weights verifying that

$$\sum_{k \in \mathcal{R}} \sum_{V_j \in Pa(G_i)} w(v_{j,t}, g_{i,k}) = 1, \forall\ pa(G_i)$$

It is interesting to note that by the way of defining how to compute the weights w we can control the bias of the individuals (bias is related to the preference of a user to one particular vote and its capability to predict the group judgements) and also the relative quality (importance) of the individuals in the group. For example, given the group in Figure 1 with $\mathcal{R} = \{1, 2, 3\}$, assuming that all the users are equal for prediction purposes and that there is no individual bias, i.e. the weights might be defined as follows

$$w(v_{j,t}, g_{i,k}) = \begin{cases} \frac{1}{|Pa(G_i)|} & \text{if } k = t, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Then, we have that $Pr(G_i|1, 2, 2, 2) = \{0.25, 0.75, 0.0\}$ and $Pr(G_i|1, 2, 2, 3) = \{0.25, 0.5, 0.25\}$.

Propagating with canonical weighted-sum. Assuming that, given the members opinion, the group's rating is independent of the $\mathcal{IS}$, the *exact* a posteriori probabilities for group nodes, $Pr(G_i = s|ev)$, can be computed efficiently by means of a straight application of the following theorem[1]:

Theorem 1. *Let G_i be a group node, let $V_1, \ldots, V_n$ be the individuals in the group. If the conditional probability distributions can be expressed with a canonical weighted sum and the evidence, ev, belongs to the information sources, $\mathcal{IS}$, then the exact a posteriori probability distribution for the group can be computed using the following formula:*

$$Pr(g_{i,s}|ev) = \sum_{j=1}^{n} \sum_{t=1}^{r} w(v_{j,t}, g_{i,s}) \cdot Pr(v_{j,t}|ev).$$

[1] Due to the lack of space, we do not include the proof of the theorems.

We consider this theorem important because it expresses that exact propagation can be done without imposing any restriction about the dependence or independence among individuals patterns of rating. Moreover, when there is no individual bias, i.e., the weights can be defined as $w(v_{j,t}, g_{i,s}) = w_j$ if $t = s$, and 0 otherwise, our model coincides with the classical Linear Opinion Pool. Thus Linear Opinion Pool can be considered as a particular case of the average gate.

3.2 Majority Gate

Our objective in this section is to model the Majority criterion where the final decision will depend on a simple counting of the votes received for each rating from the individuals. The rate which receives the largest number of votes is then selected as the consensus (majority) decision. This is the usual combination strategy when, for each individual, we only know the label representing his/her rate.

Definition 1 (Majority Gate). *A group node G_i is said that represents a majority combination criterion if given a configuration of its parents $pa(G_i)$ the conditional probability distributions can be represented as*

$$Pr(G_i = s|pa(G_i)) = \begin{cases} \frac{1}{m} & \text{if } s = \arg\max_k count(k, pa(G_i)) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $count(k, pa(G_i))$ is a function returning the number of occurrences of the state k in the configuration $pa(G_i)$, and m is the number of states where $count(k, pa(G_i))$ reaches the maximum value.

For example, considering a node G_i with five parents and with three candidate rates, ranging from 1 to 3, then $Pr(G_i|1, 1, 2, 1, 1) = (1, 0, 0)$ and $Pr(G_i|1, 2, 2, 1, 3) = (0.5, 0.5, 0)$. This representation of the majority gate implies an important saving in storage (we can compute its values when needed). Nevertheless, and in order to combine the individuals opinions, we need to perform the exponential number of computations. We shall see how these computations can be performed efficiently.

Propagating with majority gates: A key idea behind majority criterion is that the order in which the individuals are considered does not matter, and therefore there exist many different configurations collapsing to the same situation. For example, consider that four individual vote with 1 and one individual votes with 2. In this case there are five different configurations representing the same situation, i.e. $pa_1(G_i) = \{2, 1, 1, 1, 1\}$, $pa_2(G_i) = \{1, 2, 1, 1, 1\}$, $pa_3(G_i) = \{1, 1, 2, 1, 1\}$, $pa_4(G_i) = \{1, 1, 1, 2, 1\}$ and $pa_5(G_i) = \{1, 1, 1, 1, 2\}$.

It must be noticed that since the order is not a factor, we might talk about combinations. We will denote by $\Delta(G_i)$ the set of combinations with repetition from the individual votes in $Pa(G_i)$ and we use $\delta(G_i)$ or $< >$ to denote a single combination. Thus, the above situation should be represented by $\delta(G_i) =< 1, 1, 1, 1, 2 >$. Considering that the number of parents of G_i is n and that each

parent has r different states we find that the number of possible combinations with repetition is $CR_n^r = (n+r-1)!/(n!(r-1)!)$

The next theorem shows that in order to combine the different individual rates we only need to take into account the probability distributions associated to the set of combinations with repetition.

Theorem 2. *Let G_i a group node in a BN whose conditional probability distributions are represented using a majority gate, let $\Delta(G_i)$ be the set of possible combinations with repetition of the values in its parent set, $Pa(G_i)$, then*

$$Pr(G_i = s|ev) = \sum_{\delta(G_i)\in\Delta(G_i)} Pr(G_i = s|\delta(G_i))Pr(\delta(G_i)|ev) \tag{4}$$

This theorem shows that if we know $Pr(\delta(G_i)|ev)$, the combination of the information with a majority gate could be done in a time proportional to the size of CR_n^r, i.e. in the order of $O(n^{r-1})$. Taking into account that in many situations $r << n$, this implies important savings with respect to considering the number of possible configurations, $O(r^n)$. For instance, if $n = 20$ and $r = 2$ we have that $CR_n^r = 21$ whereas the number of configurations (permutations) is more than 1 million.

Nevertheless, to compute $Pr(\delta(G_i)|ev)$ we must sum over all the possible configurations in the combination, i.e.

$$Pr(\delta(G_i)|ev) = \sum_{pa(G_i)\in\delta(G_i)} Pr(pa(G_i)|ev)$$

where $pa(G_i) \in \delta(G_i)$ represents that the combination with repetition $\delta(G_i)$ can be obtained from the configuration $pa(G_i)$ by removing the order constraints. Thus, since we shall need to compute these probability values for all the possible combinations, we find that an exponential number of computations will be required to obtain the group decision.

Assuming independence to approximate $Pr(\delta(G_i)|ev)$. Since we want to compute $Pr(\delta(G_i)|ev)$ efficiently, we propose to approximate this joint distribution by assuming independence between the individuals. Although this assumption might be very restrictive, it has been demonstrated very fruitful in practical purposes when combining information [4,6].

Firstly, and with the idea of being general, we are going to introduce some notation: Let $\pi(x)$ be any configuration of n independent variables $X_1, \ldots X_n$. As these variables are independent $Pr(\pi(x)) = \prod_{i=i}^{n} Pr(x_{i,j})$, $x_{i,j}$ being the value that variable X_i takes in the configuration $\pi(x)$. Let δ_k be a combination with repetition of a subset of k variables, $X_1, \ldots X_k$, and $s \in \delta_k$ represents the fact that there exists at least one variable taking the value s in the combination δ_k. Also, we say that δ_{k-1} is a s-reduction of δ_k, denoted by $\delta_k^{\downarrow s}$, if δ_{k-1} can be

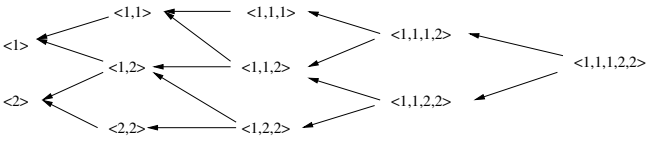


Fig. 2. Recursion graph for computing $Pr(< 1, 1, 1, 2, 2 >)$

obtained by removing a value s from the combination δ_k. The following theorem shows how $Pr(\delta_n)$ can be computed recursively:

Theorem 3. *Let δ_n be any combination with repetition from the set of $X_1, \ldots, X_n$. Then, if X_i is independent of $X_j, \forall i \neq j$, the probability associated with the combination δ_n can be computed as*

$$Pr(\delta_n) = \begin{cases} Pr(x_{1,k}) & \text{if } n = 1, \text{ with } \delta_1 =< k > \\ \sum_{s \in \delta_n} Pr(\delta_{n-1}^{\downarrow s}) Pr(x_{n,s}) & \text{if } n > 1 \end{cases} \quad (5)$$

A first idea should be to apply directly this result to compute $Pr(\delta(G_i)|ev)$. For instance, Figure 2 shows the recursion graph for the computation of $Pr(< 1, 1, 1, 2, 2 >)$, where each different combination obtained after a reduction has been displayed only once. The key observation is that the number of (sub)combinations obtained after applying a reduction process is relatively small. Thus, a recursive algorithm may encounter each one of them many times in different branches of its recursion graph. For example, Figure 2 shows that the (sub)combination $Pr(< 1, 1, 2 >)$ should be computed two times and the (sub)combination $Pr(< 1, 1 >)$ three times. Moreover, some of these subproblems might also appear when computing different joint probabilities, like $Pr(< 1, 1, 2, 2, 2 >)$. Therefore, applying directly Theorem 3 does more work than necessary.

We propose to compute every probability for a given subcombination just once and then saves its values in a table, thereby avoiding the work of recomputing this probability every time the subcombination is encountered.

The next algorithm shows how to compute the joint probability distributions for all the possible combinations with replacement in the set Δ. We follow a bottom-up approach where we first compute the probabilities associated with the (sub)combinations with lower size, being the size the number of variables used to form the combinations with repetition, and these probabilities will be used as the base of the calculus for the combinations with greater size. Initially, when considering the first variable X_1, we have r different combinations with replacement, one for each possible value of the variable X_1. Then, in a general stage, we found out that the probabilities associated with each combination δ_k of the first k variables are used in the computation of the probabilities of r different combinations with size $k + 1$, one for each possible value of the variable X_{k+1}. Each one of these combinations will be denoted by $\delta_{k \cup s}$ with $1 \leq s \leq r$.

Computing $Pr(\Delta)$

$Pr(\delta_1) = Pr(X_1)$
`for ( k = 1; k<n; k++ )`
`for each` $\delta_k \in CR_k^r$ `do // for each combination of size k`
`for (s = 1; s<=r; s++ )` `//for each value of` X_{k+1}
$Pr(\delta_{k \cup s}) += Pr(\delta_k) \times Pr(x_{k+1,s})$

An inspection of the algorithm yields a running time of $T(n) = \sum_{i=1}^{n} rCR_i^r$, i.e. $T(n) \in O(rn^r)$, being much more efficient than applying directly the recursive algorithm from Theorem 3. For example, in case of bivaluated variables, as the usual case in decision problems, we have a quadratic algorithm for combining the output of the different individuals. With respect to the memory needed to store the intermediate results we find out that the values in the stage k are only used in the stage $k-1$, therefore the used memory is in the order of $O(CR_n^r)$.

4 Experimentation: Recommending Movies for Groups

In order to study the performance of the use of AVG or MAJ gates when combining probabilistic information we consider the following problem: The prediction of the rate with which a group of people might score a given movie.

The data sets: With respect to the used datasets, they have been obtained from MovieLens[2]. Since MovieLens does not include group information we have decided to build them from the MovieLens training sets[3]. We have used two different criteria trying to capture different processes behind the creation of a group: (C1) Implementing the idea of *the group of my colleagues*, we set each user as the group administrator and we look for the 10 most similar users (those which are positively correlated with the administrator in the training dataset). Then, we select those groups of five individuals with the only restriction that at least they have rated (observed) one movie in common. Note that since similarities are not transitive, this criterion does not imply necessarily having groups with highly correlated members. (C2) Second, we have decided to fix a group (also with five individuals) with the only restriction that all the members of the group must rate at least four common movies.

With respect of the group test sets, they are obtained from each one of the MovieLens test sets. Particularly, whenever we find a movie in the test sets which has been rated by all the members of a group we include the tern *(group ID, movie ID, group rate)* in its group test dataset. Note that the *group rate* is obtained, by means of a deterministic function `CombineRate(`$r_1, \ldots, r_n$`)`, as the

[2] MovieLens was collected by the GroupLens Research Project at the University of Minnesota. The dataset contains 1682 movies and 943 users, containing 100,000 transactions where a user rates a movie using a scale from 1 to 5.

[3] With the idea of using 5 fold cross validation, we have used 5 different data subsets, each one obtained by splitting MovieLens into two disjoint sets, the first one for training (with 80% of the data) and the second one for test (with 20% of the data).

average or the majority individual's true rate, $r_1, \ldots, r_n$. Therefore, combining the decision used by a group to rate a movie and the criterion used to form a group, we obtain four different test datasets, i.e. AVG-C1 and MAJ-C1 (with a mean of 115 different records) and AVG-C2 and MAJ-C2 (with a mean of 17524 records).

Selecting a rate: Given $Pr(X)$ encoding a probability distribution over the candidate rates, the problem is to determine which is the output rate that should be recommended for X. Two basic alternatives might be considered for the `RateSelection` process:

MP Maximum a posteriori probability, i.e. rate $= \arg\max_s\{Pr(X = s|ev)\}$.
AP A posteriori average rate, i.e. rate $= \sum_{k=1}^{r} k \times Pr(X = k|ev)\}$.

Experimental framework: The objective is to predict which is the rate that a group of people will use to score an unobserved movie, I. We assume that for each member of the group, V_k, we know a probability distribution representing the belief about how this individual should rate this movie, i.e. $Pr(V_k = s|I)$. Particularly, in this experimentation, these probabilities are estimated using a *Collaborative-based RSs* [12], i.e. they have been estimated by considering the ratings that users similar to V_k have given to the movie I.

In this framework it might be possible that, for a given user V_k, none of the users similar to him had rated the movie I. In this situation the probabilities $Pr(V_k|I)$ have been estimated without information. Therefore, and in order to study the bias that the a priori distribution might produce in the predicted rate, we will consider a modification of the approaches used to select a rate: The idea is to use only the new piece of evidence that each candidate rate receives, computed as the difference between the a priori (without evidence) and the a posteriori probability values, i.e. $Pr(G_a = s|ev) - Pr(G_a = s)$. Note that this idea could be used with both AP (denoted by PD+AP) or MP (denoted by PD+MP) in the `RateSelection` process.

In order to study the performance of the combination methods, we are going to consider two different situations: In the first one, that could be considered the *Baseline* (see left hand side of table below), the predicted rate is defined as the average or majority rate of $r_1, \ldots, r_n$, being r_k the rate that, individually, each member predicts using `RateSelection`. The second alternative (right hand side below) consists on, firstly using the AVG or MAJ gates, combining the individuals probability distributions into a unique group distribution which ideally represents the group pattern of vote (`CombineProb(`$Pr_1, \ldots, Pr_n$`)`). Then, the group rate is selected.

Baseline	Using Group Layer
`For each` $V_k \in G$ `do`	$Pr(G\|ev) =$ `CombineProb(`$Pr_1, \ldots, Pr_n$`)`
$r_k =$ `RateSelection(`Pr_k`)`	`G_rate = RateSelection(` $Pr(G\|ev)$`)`
`G_rate = CombineRate(`$r_1, \ldots, r_n$`)`	

Table 1. Experimental Results

			(C1) Using Similarity				(C2) Common movies			
Group	Comb.	Rate	Baseline		Group layer		Baseline		Group layer	
Dec.	Gate	Selection	%S	MAE	%S	MAE	%S	MAE	%S	MAE
AVG	AVG	AP	47,53	0,566	45,37	0,590	57,40	0,441	58,90	0,422
		PD+AP	60,86	0,398	61,42	0,402	66,76	0,337	64,76	0,361
	MAJ	MP	45,63	0,495	62,13	0,394	44,33	0,486	62,12	0,392
		PD+MP	44,99	0,497	62,32	0,392	45,05	0,481	61,86	0,395
MAJ	AVG	AP	43,01	0,655	40,72	0,683	47,82	0,578	48,23	0,567
		PD+AP	54,64	0,476	58,24	0,446	56,30	0,457	58,44	0,438
	MAJ	MP	59,34	0,461	60,10	0,424	55,11	0,493	58,05	0,446
		PD+MP	59,19	0,461	60,29	0,422	55,72	0,486	58,35	0,443

Two different accuracy measures will be considered [13]: the percentage of success (%S), which measures the frequency with which the system makes correct predictions and the mean absolute error (MAE), which measures the average absolute deviation between a predicted rate and the group's true rate.

Table 1 presents the average results obtained after repeating the experiment with the 5 folds. First column represents the criterion used in the test dataset to decide the group rate. Second column represents the canonical model used in the group layer. In this experimentation, we have used an unbiased uniform weighting scheme in the AVG gate, i.e. $w(v_{j,s}, g_{i,k}) = 1/|Pa(G_i)|$ if $k = s$ and 0 in the other cases. Note that this model corresponds with the classical Linear Opinion Pool when all the users are considered equivalent for prediction purposes. Third column represents the criteria used in the `RateSelection` process. The next two columns represents how the groups have been constructed. In this table, the results for each particular dataset can be found in the cells indexed by the pair (*Group Dec., group construction criterion*).

From this table, as general conclusions, we can say that: i) it is better to use BN (the AVG or MAJ gates) to combine individual preferences, ii) MAJ gate could be preferable in the case it is unknown how the group (true) decisions are obtained (quite good results have been obtained using also MAJ gates when the *"real"* group vote is obtained using AVG), iii) With respect to the AVG gate, it seems preferable to use PD+AP to correct the a priori bias of the AVG gate. Also, better results have been obtained when the groups have been formed without considering similarities (we believe that these results can be improved by using proper weights in AVG gate), iv) With respect to MAJ gate it seems to perform better when considering similar users. No significant differences can be found between the use of MP or PD+MP.

5 Conclusions

A general BN-based model for combine probabilistic information in a group recommending framework has been proposed in this paper. With this model the

interaction among the individuals when deciding a group rate are represented intuitively by means of the use of AVG (average) and MAJ (majority) canonical modes. Linear time inference algorithms (assuming independence in the case of MAJ gate) have been developed to compute the a posteriori distribution for the group. These distributions represent the group preferences for a given item. Experimental results show the validity of our proposal.

By way of future work, we are planning to evaluate the model with real data, involving real groups to determine the quality of the recommendations provided and also to apply these methodology to problems as the combination of classifiers.

Acknowledgements

This work has been supported by the Spanish 'Ministerio de Educación y Ciencia' and 'Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía' under Projects TIN2005-02516 and TIC-276, respectively.

References

1. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: A recommender system for groups of user. In: Proceedings of the Seventh European Conference on Computer-Supported Cooperative Work, pp. 199–218 (2001)
2. McCarthy, J.E., Anagnost, T.D.: Musicfx: an arbiter of group preferences for computer supported collaborative workouts. In: CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work, p. 348. ACM Press, New York (2000)
3. Pennock, D.M., Wellman, M.P.: Graphical models for groups: Belief aggregation and risk sharing. Decision Analysis 2(3), 148–164 (2005)
4. Clemen, R.T., Winkler, R.L.: Combining probability distributions from experts in risk analysis. Risk Analysis 19, 187–203 (1999)
5. Genest, C., Zidek, J.: Combining probability distributions: A critique and annotated bibliography. Statistical Sciences 1(1), 114–148 (1986)
6. Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifiers. IEEE Trans. on Pattern Anal. and Machine Intell. 20(3), 226–238 (1998)
7. Chen, Y., Chao-Hsien Chu, T.M., Pennock, D.M.: Information markets vs. opinion pools: An empirical comparison. In: EC '05: Proceedings of the 6th ACM conference on Electronic Commerce, pp. 58–67. ACM Press, New York (2005)
8. Druzdzel, M., Diez, F.J.: Combining knowledge from different sources in causal probabilistic models. Journal of Machine Learnig Research 4, 295–316 (2003)
9. McCarthy, K., Salam, M., Coyle, L., McGinty, L., Smyth, B., Nixon, P.: Group recommender systems: a critiquing based approach. In: IUI '06: Proceedings of the 11th international conference on Intelligent user interfaces, pp. 267–269. ACM Press, New York (2006)
10. Masthoff, J., Gatt, A.: In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. User Model User-Adap Inter. 16, 281–319 (2006)

11. Yu, Z., Zhou, X., Hao, Y., Gu, J.: Tv program recommendation for multiple viewers based on user profile merging. User Model. and User-Adap. Inter. 16(1), 63–82 (2006)
12. de Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Rueda-Morales, M.A.: Group recommending: A methodological approach based on bayesian networks. In: IEEE-Workshop on Web Personalization, Recommender Systems and Int. User Interface, pp. 835–845. IEEE Computer Society Press, Los Alamitos (2007)
13. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22(1), 5–53 (2004)
14. de Campos, L.M., Fernández-Luna, J.M., Huete, J.F.: The bnr model: foundations and performance of a bayesian network-based retrieval model. Int. J. Approx. Reasoning 34(2-3), 265–285 (2003)

A Fast Hill-Climbing Algorithm for Bayesian Networks Structure Learning

José A. Gámez, Juan L. Mateo, and José M. Puerta

Computing System Department and SIMD-$i^3\mathcal{A}$
University of Castilla-La Mancha
02071, Albacete, Spain
{jgamez,juanlmc,jpuerta}@dsi.uclm.es

Abstract. In the score plus search based Bayesian networks structure learning approach, the most used method is hill climbing (HC), because its implementation is good trade-off between CPU requirements, accuracy of the obtained model, and ease of implementation. Because of these features and to the fact that HC with the classical operators guarantees to obtain a minimal I-map, this approach is really appropriate to deal with high dimensional domains. In this paper we revisited a previously developed HC algorithm (termed *constrained* HC, or CHC in short) that takes advantage of some scoring metrics properties in order to restrict during the search the parent set of each node. The main drawback of CHC is that there is no warranty of obtaining a minimal I-map, and so the algorithm includes a second stage in which an unconstrained HC is launched by taking as initial solution the one returned by the constrained search stage. In this paper we modify CHC in order to guarantee that its output is a minimal I-map and so the second stage is not needed. In this way we save a considerable amount of CPU time, making the algorithm best suited for high dimensional datasets. A proof is provided about the minimal I-map condition of the returned network, and also computational experiments are reported to show the gain with respect to CPU requirements.

1 Introduction

Data mining goal can be understood as compressing the available data into a more compact representation called *model*. Later this model can be used to tackle different descriptive (e.g. identifying dependences relations, clusters, etc.) or predictive (e.g. classification, computing posterior beliefs) tasks. *Bayesian Networks* (BNs) [20,16,17] have become one of the favorite knowledge representation formalisms for model-based data mining because of their double descriptive/predictive capability and their innate uncertainty management.

Bayesian Networks (BNs) are graphical models able to represent and manipulate efficiently n-dimensional probability distributions [20]. The knowledge base a BN encodes can be viewed as a double representation model divided into a qualitative (a directed acyclic graph or dag) and a quantitative part (a set of

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 585–597, 2007.

locally specified probability distributions). Thus, descriptive tasks are carried out by performing *relevance analysis* over the graph, while predictive tasks are based on a clever use of the (in)dependences codified in the dag to allow efficient probabilistic inference. Being BNs so attractive models and due to the increasing availability of data, it is not strange the large number of works found in the literature to tackle with the BN structure learning problem. Somewhat generalizing, there are two main approaches for learning BNs:

• *Score+search methods.* A function f is used to score a network/dag with respect to the training data, and a search method is used to look for the network with best score. Different scoring metrics (Bayesian and non-Bayesian [17, ch. 8][14]) and search methods (mainly of heuristic nature because the NP-hardness of BN structure learning problem [9]) have been proposed.
• *Constraint-based methods.* The idea underlying these methods is to satisfy as many independences present in the data as possible [21][17, ch. 10]. Statistical hypotheses testing is used to determine the validity of conditional independence sentences. There also exist hybrid algorithms that combine these two approaches, e.g. [2] or even hybrid scoring metrics [4].

In BN structure learning in the space of dags, local search-based approach ([14,8,6,7,13][17, ch. 9]) is the outstanding one when dealing with large datasets (specially large number of variables), because its good trade-off between resources required (e.g. CPU time) and accuracy of the obtained model.

In this work we focus on local search methods and concretely in one of its simpler versions, hill climbing, with the goal of showing how its efficiency (CPU time requirements) can be reduced without decreasing their accuracy. Thus, we follow the research line started in [15] where a *constrained hill climbing* (CHC) algorithm for BN structure learning was proposed. In [15] we took advantage during the local search of some interesting properties of scoring metrics, to remove as candidate parents (of X) those nodes found as conditionally independent given the current parent set. As we do this at each step of CHC the number of statistics to be computed significantly decreases and so CHC is faster than pure HC. However, this way of proceeding can finish in a suboptimal solution, so the last step of CHC is to launch an unconstrained HC by taking the solution previously found as starting point. Here, we propose a new version of CHC that avoids the last step, resulting in a faster one. We provide theoretical results about the type of networks obtained and also an experimental comparison with CHC and HC.

The paper is structured as follows: We begin in Section 2 with some preliminaries about Bayesian networks and local search in the space of graphs. Sections 3 and 4 constitute the core of this paper, because in them we develop the algorithm we propose and experimentally evaluate it. Finally, in Section 5 we present our final conclusions and outline future research.

2 Preliminaries

In this section we set the notation and review basics concepts about BNs and BNs structural learning by using local search.

2.1 Bayesian Networks Basics

A Bayesian Network (BN) [20] is a tuple $\mathcal{B} = (G, \mathbf{P})$, where: (a) $G = (\boldsymbol{V}, E)$ is a dag whose nodes $\boldsymbol{V} = \{X_1, X_2, \ldots, X_n\}$ represent the problem variables and whose topology (arcs in E) encodes conditional (in)dependence relationships among the variables; and (b) $\mathbf{P}$ is a set of locally specified conditional probability distributions $\{P(X_i|pa_G(X_i)\ :\ i = 1, \ldots, n\}$, $pa_G(X_i)$ being the parent set of X_i in G. From these conditional distributions we can recover the joint distribution over $\boldsymbol{V}$:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|pa_G(X_i)) \tag{1}$$

We denote that variables in $\mathbf{X}$ are conditionally independent (through d-separation) of variables in $\mathbf{Y}$ given the set $\mathbf{Z}$, in a dag G as $\langle \mathbf{X}, \mathbf{Y}|\mathbf{Z}\rangle_G$. The same sentence but in a probability distribution p is denoted as $I_p(\mathbf{X}, \mathbf{Y}|\mathbf{Z})$. A dag G is an *I-map* of a probability distribution p if $\langle \mathbf{X}, \mathbf{Y}|\mathbf{Z}\rangle_G \Longrightarrow I_p(\mathbf{X}, \mathbf{Y}|\mathbf{Z})$, and is *minimal* if no arc can be eliminated from G without violating the I-map condition. G is a *D-map* of p if $\langle \mathbf{X}, \mathbf{Y}|\mathbf{Z}\rangle_G \Longleftarrow I_p(\mathbf{X}, \mathbf{Y}|\mathbf{Z})$. When a dag G is both an I-map and a D-map of p, it is said that G and p are *isomorphic* models. It is always possible to build a minimal I-map of any given probability distribution p, but some distributions do not admit an isomorphic model [20]. In general, when learning Bayesian networks from data our goal is to obtain a dag being a minimal I-map of the probability distribution encoded by the dataset.

2.2 Learning BNs by Local Search

The problem of learning the structure of a BN can be stated as follows: Given a training dataset $D = \{\mathbf{v}^1, \ldots, \mathbf{v}^m\}$ of instances (configurations of values) of $\boldsymbol{V}$, find the dag G^* such that

$$G^* = \arg\max_{G \in \mathcal{G}_n} f(G : D) \tag{2}$$

where $f(G : D)$ is a scoring metric which evaluates the merit of any candidate dag G with respect to the dataset D, and $\mathcal{G}_n$ is the set containing all the dags with n nodes.

Local Search (concretely Hill-Climbing) methods traverse the search space starting in an initial solution and doing a finite number of steps. At each step the algorithm only considers local changes, i.e. neighbor dags, and chooses that resulting in the greatest improvement of f. The algorithm stops when there is no local change yielding an improvement of f. Because of this greedy behavior the execution stops when the algorithm is trapped in a solution that most times locally maximizes f rather than globally maximizing it. Different strategies are used to try to escape from local optima: restarts, randomness, etc.

The effectiveness and efficiency of a local search procedure depends on several aspects, like the neighborhood structure considered, the starting solution or the ability of fast evaluation of candidate subgraphs (neighbors). The neighborhood structure considered is directly related with the operators used to generate

neighbors by applying local changes. In BN learning, the usual choices for local changes in the space of dags are arc addition, arc deletion and arc reversal. Of course, except in arc deletion we have to take care of avoiding to introduce directed cycles in the graph. Thus, there are $O(n^2)$ possible changes, n being the number of variables. With respect to the starting solution, the empty network is usually considered although random starting points or perturbed local optima are also used, specially in the case of iterated local search.

Efficient evaluation of neighbors/dags is based on an important property of scoring metrics: *decomposability* in presence of full data. In the case of BNs decomposable metrics evaluate a given dag as the addition of its nodes family score, i.e., the subgraphs formed by a node and its parents in G. Formally, if f is decomposable then:

$$f(G:D) = \sum_{i=1}^{n} f_D(X_i, Pa_G(X_i)) \tag{3}$$

$$f_D(X_i, Pa_G(X_i)) = f_D(X_i, Pa_G(X_i) : N_{x_i, pa_G(X_i)}) \tag{4}$$

where $N_{x_i,pa_G(X_i)}$ are the statistics of the variables X_i and $Pa_G(X_i)$ in D, i.e, the number of instances in D that match each possible instantiation of X_i and $Pa(X_i)$.

Thus, if a decomposable metric is used, a procedure that changes only one arc at each move can efficiently evaluate the neighbor obtained by this change. This kind of (local) methods can reuse the computations carried out at previous stages, and only the statistics corresponding to the variables whose parents has been modified need to be recomputed. It is clear that a hill climbing algorithm using the operators above described can take advantage of this operation mode, concretely it has to measure the following differences when evaluating the improvement obtained by a neighbor dag:

1. Addition of $X_j \rightarrow X_i$: $f_D(X_i, Pa_G(X_i) \cup \{X_j\}) - f_D(X_i, Pa_G(X_i))$
2. Deletion of $X_j \rightarrow X_i$: $f_D(X_i, Pa_G(X_i) \setminus \{X_j\}) - f_D(X_i, Pa_G(X_i))$
3. Reversal of $X_j \rightarrow X_i$: It is obtained as the sequence: deletion$(X_j \rightarrow X_i)$ plus addition$(X_i \rightarrow X_j)$, so we compute $[f_D(X_i, Pa_G(X_i) \setminus \{X_j\})$ $-f_D(X_i, Pa_G(X_i))] + [f_D(X_j, Pa_G(X_j) \cup \{X_i\}) - f_D(X_j, Pa_G(X_j))]$

3 Fast-CHC: An One Iteration Constrained Hill Climbing for Learning BNs

Apart from decomposability there are other desirable properties of scoring metrics. Among them the following one was the key point in the design of CHC [15] algorithm:

Definition 1. *Let D be a dataset containing m iid samples from some distribution p. Let G be any dag, and G' the dag obtained by adding edge $X_i \rightarrow X_j$ to G. A scoring metric is* locally consistent *if in the limit as m grows large, the following two conditions hold:*

1. *If* $\neg I_p(X_i, X_j|Pa_G(X_j))$, *then* $f(G:D) < f(G':D)$
2. *If* $I_p(X_i, X_j|Pa_G(X_j))$, *then* $f(G:D) > f(G':D)$

Usually considered scoring metrics as BDe, MDL and BIC are locally consistent [8]. From Definition 1 it can (asymptotically) be assumed that the differences computed by a locally consistent scoring metric f can be used as conditional independence tests over the dataset D. To do this, it is enough to suppose that D constitutes a sample which is isomorphic or that the distribution p is faithfull to a DAG, that is, a perfect-map of p can be obtained via D. In fact, CHC [15] is not the only approach to learn BNs in which the difference between scores is used as a conditional independence test, thus in [1,19,18] Bayesian metrics have been used to perform conditional tests finding that, in general, they are more reliable than traditional chi-square based conditional test.

CHC algorithm [15] can be viewed as a two-iterations algorithm: in the first one it takes advantage of this property to constrain the number of modifications (neighbors) to be explored at each iteration of the hill climbing procedure; and, in the second one an unconstrained hill climbing is launched by taking as starting point the solution obtained in the first phase. This second iteration is necessary to ensure that the solution finally returned is an I-map. Table 1 shows the algorithm CHC (and Fast-CHC).

As we can see CHC (and Fast-CHC) restricts the neighborhood of a dag G by constraining the set of allowed parents for each node X_i. To do this, we associate a set of *forbidden parents* (FP) to each node. The content of $FP(X_i)$ is modified by using the information provided by the differences computed at the current step, concretely we use definition 1 (locally consistent metric) to update $FP(X_i)$:

- Addition. If when adding $X_j \rightarrow X_i$ we get $d < 0$, then (asymptotically) $I_p(X_i, X_j|Pa_G(X_i))$, and so we do not have to test[1] anymore the addition of X_j as parent of X_i.
- Deletion. This case is analogous to addition. Now, if we get $d > 0$ when deleting $X_j \rightarrow X_i$, then again we have that (asymptotically) $I_p(X_i, X_j|Pa_G(X_i))$.

Here, we have used metric differences as a sort of conditional independence tests, but in a more general framework, we could use any conditional independence test to manage the FP set. In our work we use metric differences to get reliable test and to save CPU time because all the necessary scores to do the test have been previously computed.

The main goal of this work is to investigate how to avoid the second HC iteration, because it needs a considerable amount of CPU time which degrades the whole performance of CHC algorithm. However, to avoid this second iteration we must guarantee that the dag obtained in the first one is an I-map. To do this we have to introduce some changes into CHC algorithm, obtaining a new version

[1] In fact there are two versions of CHC, one based on forbidding the arc, i.e., X_j as parent of X_i, and other base on forbiding the undirected edge, i.e., X_j as parent of X_i but also X_i as parent of X_j.

Table 1. Algorithm(s) for learning Bayesian networks by using hill climbing-based approaches. Hill climbing is described in normal text, while if underlined text is added we get CHC algorithm. Finally in step 5 we add the modifications needed to get Fast-CHC (marked in boldface and preceded by [Fast-CHC]).

Algorithm Fast-CHC
1. Initialization: Choose a dag G as the starting point. $\forall X_i$ do $FP(X_i) = \emptyset$.
2. Neighbors generated by addition: For every node X_i and every node $X_j \notin (Pa_G(X_i) \cup FP(X_i))$ compute the difference d between G and $G \cup \{X_j \rightarrow X_i\}$ as described in Section 2.2. If $d < 0$ then add $\{X_j\}$ to $FP(X_i)$ and add $\{X_i\}$ to $FP(X_j)$. Of course, neighbors in which adding $X_j \rightarrow X_i$ induces a directed cycle are not taken into account. Store the change which maximizes d.
3. Neighbors generated by deletion: For every node X_i and every node $X_j \in Pa_G(X_i)$ compute the difference d between G and $G \setminus \{X_j \rightarrow X_i\}$ as described in Section 2.2. If $d > 0$ then add $\{X_j\}$ to $FP(X_i)$ and add $\{X_i\}$ to $FP(X_j)$. Store the change which maximizes d.
4. Neighbors generated by reversal: For every node X_i and every node $X_j \in Pa_G(X_i)$, such that, $X_i \notin FP(X_j)$ compute the difference d between G and $G \setminus \{X_j \rightarrow X_i\} \cup \{X_i \rightarrow X_j\}$ as described in Section 2.2. In this case $d = d_1 + d_2$, where d_1 is the difference obtained by removing X_j as parent of X_i, and d_2 is the difference obtained by adding X_i as parent of X_j. If $d_1 > 0$ or $d_2 < 0$ then add $\{X_j\}$ to $FP(X_i)$ and add $\{X_i\}$ to $FP(X_j)$. Again, modifications inducing directed cycles are avoided. Store the change maximizing d.
5. From the three changes stored in the previous steps takes the one which maximizes d. If $d \leq 0$ then stops the algorithm and - [HC,Fast-CHC] return G - [CHC] return HC(G) else: - [HC,CHC] modify G by applying the selected change and return to step 2 - [**Fast-CHC**] modify G by applying the selected change and in the case of adding $X_j \rightarrow X_i$, then **re-compute** $FP(X_i)$ **and** $FP(X_j)$ **as** $FP(X_i) = FP(X_i) \setminus Adj_G(X_j)$ $\forall X_a \in Adj_G(X_j)$**, them re-compute** $FP(X_a) = FP(X_a) \setminus \{X_i\}$ **and** $FP(X_j) = FP(X_j) \setminus Adj_G(X_i)$**.** $\forall X_b \in Adj_G(X_i)$**, them re-compute** $FP(X_b) = FP(X_b) \setminus \{X_j\}$ Return to step 2.

that we call *Fast Constrained Hill Climbing* (Fast-CHC), shown in Table 1 and discussed in the next paragraphs.

First, let us discuss about two basic points: monotonicity and termination. We can easily see that Fast-CHC is monotone with respect to f, i.e., $f(G : D) \leq f(G' : D)$, where G' is the neighbor of G which maximizes the difference d. From this monotonic behavior and due to the fact that Fast-CHC stops when there is no neighbor of G which improves $f(G)$ termination is guaranteed.

Now, let us discuss about the two more important properties of HC-based BNs learning algorithms: (1) HC algorithms are *efficient* because of their local behavior, and (2) under faithfulness conditions and no matter which starting point is used, they guarantee to obtain an I-map minimal of p [15]. Because of these two properties HC algorithms are very good candidates to be used in high dimensional databases, and so, it is very interesting for us to contribute with new improvements that try to speedup HC algorithms but maintaining at the same time the quality of their output. In [15] we demonstrate that an algorithm that *only* restricts the parents set by using the subset $FP()$ could get stuck in a locally sub-optimal solution, and so we introduce a second (unconstrained) HC iteration to ensure the I-map condition. Here, CHC algorithm is updated to Fast-CHC by (1) re-computing $FP()$ subsets when addition is the selected operator, and (2) removing the second HC iteration. Concretely, when addition of $X_j \rightarrow X_i$ is the selected movement, then we remove any node adjacent to X_j from $FP(X_i)$ and any node adjacent to X_i from $FP(X_j)$, beside we also remove the inverse direction, i. e., if we remove X_a from $FP(X_b)$ then we remove X_b from $FP(X_a)$.

The following proposition proves the type of output provided by algorithm Fast-CHC.

Proposition 1. *Let D be a dataset containing m iid samples from some distribution p. Let $\hat{G}$ be the dag obtained by running Fast-CHC algorithm by taking a dag G_0 as the initial solution, i.e., $\hat{G} =$ Fast-CHC(G_0). If the metric f used to evaluate dags in Fast-CHC is locally consistent, then under the faithfulness assumption $\hat{G}$ is a minimal I-map of p in the limit as m grows large.*

Proof sketch: Because of the faithfulness assumption there exists dag G^* being a perfect-map of p, where each variable X_i is conditionally independent of the rest given their Markov Blanket, i.e., $\forall X_i$ and $\forall X_j \notin MB(X_i)$, $I_p(X_i, X_j | MB_{G^*}(X_i))$, where $MB_{G^*}(X_i) = Pa_{G^*}(X_i) \cup Ch_{G^*}(X_i) \cup Pa_{G^*}(Ch_{G^*}(X_i))$. Thus, our first step is to prove that $\hat{G}$ is an I-map of G^*, which reduces to prove that in $\hat{G}$ we have a superset of $MB_{G^*}(X_i)$ for each X_i, i. e., $MB_{G^*}(X_i) \subseteq MB_{\hat{G}}(X_i)$. Our second step will be to prove that this I-map is also minimal.

[(1) $\hat{G}$ is an I-map of G^*] First at all, we have to note that if an adjacency $X_i - X_j$ is in G^* then it is also in $\hat{G}$ because there is not subset S of variables, such that, $I_p(X_i, X_j | S)$. Therefore, $MB_{\hat{G}}(X_i)$ includes $Pa_{G^*}(X_i) \cup Ch_{G^*}(X_i)$. Now we prove that $Pa_{G^*}(Ch_{G^*}(X_i))$ is also included in $MB_{\hat{G}}(X_i)$. The three following cases must be considered:

• Case a) Let us suppose that $X_i \rightarrow X_j \leftarrow X_{p_j}$ is in $\hat{G}$ and in G^*, then by definition of MB X_{p_j} is included in $MB_{\hat{G}}(X_i)$. • Case b) Let us suppose $X_i \rightarrow X_j \rightarrow X_{p_j}$ is in $\hat{G}$ and $X_i \rightarrow X_j \leftarrow X_{p_j}$ is in G^*. Then, according to our data we should have $\neg I_p(X_i, X_{p_j} | Pa_{\hat{G}}(X_{p_j}))$ and by def. 1 we obtain $f_D(X_{p_j}, Pa_G(X_{p_j}) \cup \{X_i\}) - f_D(X_{p_j}, Pa_G(X_{p_j})) > 0$, thus we must have the arc $X_i \rightarrow X_{p_j}$ included in $\hat{G}$. Notice that X_i does not belong to $FP(X_{p_j})$ due to the step of Fast-CHC that when the arc $X_j \rightarrow X_{p_j}$ is included in $\hat{G}$ then we remove X_i from

$FP(X_{p_j})$, if it is already included, besides if $X_i \rightarrow X_j$ is included in a later step, then again X_i is removed from $FP(X_{p_j})$ due to in the last step of Fast-CHC when we remove the "indirected" link from $FP()$. Therefore $X_{p_j} \in MB_{\hat{G}}(X_i)$.
• Case c) Let us suppose $X_i \rightarrow X_j$ is in G^* and $X_i \leftarrow X_j$ is in $\hat{G}$. Suppose also that $X_i \leftarrow X_j - X_{p_j}$ is in $\hat{G}$ (it is the same case in any orientation, the reasoning is symmetrical) and $X_i \rightarrow X_j \leftarrow X_{p_j}$ is in G^*. Then, according to our data we should have $\neg I_p(X_i, X_{p_j} | Pa_{\hat{G}}(X_i))$ and applying definition 1 we obtain $f_D(X_i, Pa_G(X_i) \cup \{X_{p_j}\}) - f_D(X_i, Pa_G(X_i)) > 0$, thus we must have the arc $X_i \leftarrow X_{p_j}$ or $X_i \rightarrow X_{p_j}$ included in $\hat{G}$. Notice that X_{p_j} does not belong to $FP(X_i)$ due to the step of Fast-CHC that when the arc $X_j \rightarrow X_i$ is included in $\hat{G}$ then we remove X_{p_j} from $FP(X_i)$, and simirlarly X_i is removed from $FP(X_{p_j})$, if they are already included, therefore in any case $X_{p_j} \in MB_{\hat{G}}(X_i)$.
[(2) $\hat{G}$ is a minimal I-map of G^*] To prove the minimal condition let us suppose the converse. Then, because of the Markov Condition there exists $X_j \in Pa_{\hat{G}}(X_i)$ such that $I_p(X_i, X_j | Pa_{\hat{G}}(X_i))$. If so, $\hat{G}$ cannot be a local optimum because there is (at least) a deletion operation with positive difference. □

To end with this section we briefly discuss about related work. Thus, the underlying idea in CHC [15] and Fast-CHC has been previously used *heuristically* in the sparse candidate algorithm [13]. However, the operation mode of that algorithm is far enough of our proposal because the sparse candidate is in fact an iterated HC that at each (outer) iteration restricts the number of candidate parents for each variable X_i to the k most promising ones, k being the same value for every variable. Another important issue is that the parameter k must be specified and, in general, it will be dependent of the problem being considered. The sparse candidate algorithm can be viewed as a general framework where we can use any search algorithm in each outer iteration and, in general, the I-map condition in its output is not guaranteed.

Another algorithm very related to this work is the so-called MaxMin Hill-climbing [22]. This is a two iterations algorithm that in its first stage tries to recover the Markov Blanket for each variable while in the second stage a HC algorithm is launched but restricted to the set of adjacencies previously found. The MaxMin algorithm output, in general, is not a minimal I-map minimal.

A different approach [10] consist of obtaining a graph skeleton based on conditional independece test of order 0 and 1, and later an evolutionary algorithm is run but constrained to used only the edges in the skeleton instead of the complete graph. This algorithm is highly CPU demanding and again the I-map minimal condition is not guaranteed.

Finally, our approach is also related to classical constraint-based BN learning algorithm: PC [21], because it uses conditional independence tests $I(X, Y|Z)$ to guide the search and some of its modifications as [1] where score differences are used as conditional independences test. However, the main difference between such algorithms and our approach relies in the fact that we set the current parents as a d-separator set in G, while like-PC algorithms need to perform tests with respect to all the possible subsets of $\text{adjacent}_G(X) \setminus \{Y\}$.

4 Experimental Results

In this section we evaluate experimentally the proposed algorithm. We test Fast-CHC in two different scenarios, the first one is for modeling or representing several domains as Bayesian networks, for this task we start with a database of cases and an algorithm for learning Bayesian networks is used to model the conditional independences that the data present. In this case the performance criteria could be the better the found networks model the data the better is the algorithm used; the measures we have used are two: the BDeu score of the returned network; and (a transformation of) the Kullback-Leibler (KL) divergence of the resulting networks with respect to the data used as training set. As it is described in [6], the larger is the KL value the better is the network. BDeu (Bayesian Dirichlet equivalent with uniform priors) with equivalent sample sample equal to 1 have been used during the search and so to score the obtained networks. Apart from these quality measures, we have take into account the efficiency (CPU time and number of statistics computed) in the learning process. CPU time is shown relative to fast-CHC, that is, we use value 1 for CHC and then a value of l means that a give algorithm needs l times the CPU-time used by fastCHC to learnt the model. With respect to statistics we use a cache in order to only count the number of *different* statistics computed.

As a second scenario, we have considered a predictive task instead of the descriptive one, thus a typical use of Bayesian networks is to use them as classifiers. We focus on naive-Bayes (BN) augmented classifiers [12], where a NB structure is used as initial solution and then a search process (HC, CHC and fast-CHC) is launched but constrained to not modifying arcs related to the class variable. This type of classifiers are known as Bayesian networks Augmented Naive Bayes classifiers (BAN). In our evaluation again we attend to two parameters: quality of the resulting network and efficiency. Quality is measure by using accuracy and Brier score, and computed by using a 5x2 cross validation process. For efficiency, again we use CPU-time and different statistics computed.

All the experiments have been carried out in the same computer (Pentium IV with 3GHz, 2Gb of RAM memory and 250Gb of hard disk) in order to fairly compare reported CPU-time. All the algorithms have been written in Java using as framework the Elvira software [11].

The datasets used in the experiments come from different sources. Due to space restrictions we cannot show here details about them, but we show this information in [http://www.dsi.uclm.es/simd/experiments/chc/datasets.html].

In table 2 we show the results we have obtained when modelling data by using Bayesian networks. We can see that the quality values for Fast-CHC algorithm are always lower but are close to those yield by HC and CHC algorithms,except for sheep database. This result is not somewhat surprisingly because of we are considering in our experiments databases of real domains in several cases and in all the cases databases of limited size. Nonetheless, if we attend to efficiency results we can see that statistics calculated by Fast-CHC are reduced in a 68.2% compared to HC and in a 56.9% compared to CHC for the clean1 database. This is a great reduction, even more if we take into account that this two

Table 2. Experimental results for the algorithms tested on learning Bayesian networks

database	algorithm	Time	KL div.	Score	Stat.
alarm	HC	3.90	9.17	-54891	3498
	CHC	2.41	9.25	-54864	2286
	Fast-CHC	1	9.17	-55215	1175
clean1	HC	6.82	44.47	-12079	67557
	CHC	5.09	44.96	-12106	49837
	Fast-CHC	1	42.54	-12570	21459
farmer101	HC	8.78	28.66	-147750	28471
	CHC	20.24	28.12	-148995	22904
	Fast-CHC	1	26.37	-151112	10762
insurance	HC	2.42	8.42	-133263	1902
	CHC	1.84	8.39	-133401	1458
	Fast-CHC	1	8.31	-134035	960
lrs	HC	3.22	80.70	-29928	21820
	CHC	2.73	80.23	-30768	21117
	Fast-CHC	1	77.29	-31306	13758
sheep	HC	8.23	10.97	-199829	1341
	CHC	4.19	10.63	-202443	940
	Fast-CHC	1	10.59	-202391	627

Table 3. Results of our experiments for the algorithms tested in a classification task

database	algorithm	Time	Brier	Accuracy	KL div.	Score	Stat.
farmer50	HC	57.97	0.60	0.66	17.06	-53595	5025
	CHC	40.81	0.60	0.66	17.02	-53840	3628
	Fast-CHC	1	0.60	0.66	16.56	-54141	1527
kr-vs-kp	HC	3.36	0.07	0.96	4.94	-18045	4707
	CHC	2.43	0.07	0.96	4.94	-18060	3194
	Fast-CHC	1	0.08	0.95	4.67	-18260	1791
optdigits	HC	4.32	0.10	0.94	21.62	-164164	5930
	CHC	3.38	0.10	0.94	21.49	-164589	4769
	Fast-CHC	1	0.10	0.94	20.24	-165443	1800
sheep	BAN	189.45	0.47	0.67	8.80	-116296	958
	CHC	198.70	0.48	0.67	9.04	-116189	709
	Fast-CHC	1	0.48	0.67	8.76	-117103	329
soybean	BAN	23.01	0.11	0.92	14.57	-6043	2967
	CHC	33.66	0.11	0.92	14.57	-6047	2328
	Fast-CHC	1	0.11	0.92	14.28	-6100	803

algorithms (HC and CHC) take more benefit from the cached statistics because they perform more search steps and it is more likely that a computed score was needed later. The smallest reduction with respect to HC is 36.9% and 34.8% for CHC. Furthermore, if we look at the time needed to learn the networks we can see than the reduction range is 88.6% to 58.7% with respect to HC and 95% to 45.7% with respect to CHC. Therefore, the advantage of fast-CHC for really large domains is clear.

In the case of classifiers the efficiency results are even better. The reduction in time for Fast-CHC can reach 99% with respect both other algorithms, and the lower limit is higher too: 70.2% for HC and 58.8% for CHC. This advantage is even better when we look to number of computed statistics, where we got reduction rates which ranges from 72.9% to 62% with respect HC and from 65.5%

to 43.9% with respect to CHC. If we now pay attention to quality results, we get the same conclusion than in the descriptive task with respect to network score and KL. However, when classification parameters are considered as accuracy and Brier score, Fast-CHC compares better with respect to HC and CHC. Thus, there is only a case in which it gets slightly worse accuracy than (C)HC (kr-vs-kp), and two cases in which it gets slightly worse Brier score than HC.

5 Concluding Remarks

In this paper we have proposed a new version of constrained hill-climbing search algorithm for learning Bayesian network structure. The method consists in the use of a hill climbing algorithm with the classical operators (addition, deletion and reversal) but in which we restrict the neighborhood by using a list of parents not allowable for each variable. The main novelty here is that at each iteration the forbidden parents sets are modified in order to theoretically ensure that the output of the proposed algorithm is a minimal I-map. The underlying idea of the method relies on the theoretical property of *local consistency* exhibited by some scoring metrics as BDe, MDL and BIC. Apart from proving that the new version always returns a minimal I-map, we have conducted computational experiments in order to analyse the savings with respect to CPU time achieved by this new CHC version. Furthermore, experiments have been designed in two different tasks: (1) descriptive modelling of a data set by using a Bayesian network, and (2) predictive behaviour of a BAN classifier learnt by using CHC. We have reported quality and efficiency data about all the experiments. With respect to quality the models learnt by using fast-CHC are slightly worse than those learnt by using HC and CHC, but the difference is remarkably smallest when classification statistics (accuracy and Brier score) are used. With respect to efficiency the advantage of fast-CHC is clear.

Our main conclusion is that when the domain complexity grows, fast-CHC can be of great utility because it can obtain a Bayesian network model quickly, perhaps less accurate than the one obtained by using CH or CHC if they have enough time to return any.

For future research, we plan to carry out a more systematic experimentation and extend the comparative analysis to other related approaches. Furthermore, different local search methods and search spaces (as PDAGs [8] and RPDAGs [3]) will be considered.

Acknowledgements

This work has been partially supported by Spanish Ministerio de Educación y Ciencia, Junta de Comunidades de Castilla La Mancha and FEDER funds under projects TIN2004-06204-C03-03 and PBI-05-022.

References

1. Abellán, J., Gomez-Olmedo, M., Moral, S.: Some Variations on the PC Algorithm. In: Proc of The third European Workshop on Probabilistic Graphical Models (2006)
2. Acid, S., de Campos, L.M.: A hybrid methodology for learning belief networks: Benedict. International. Journal of Approximate Reasoning 27(3), 235–262 (2001)
3. Acid, S., de Campos, L.M.: Searching for Bayesian Network Structures in the Space of Restricted Acyclic Partially Directed Graphs. Journal of Artificial Intelligence Research 18, 445–490 (2003)
4. de Campos, L.M.: A Scoring Function for Learning Bayesian Networks based on Mutual Information and Conditional Independence Tests. Journal of Machine Learning Research 7, 2149–2187 (2006)
5. de Campos, L.M., Fernández-Luna, J.M., Gámez, J.A., Puerta, J.M.: Ant colony optimization for learning Bayesian networks. International Journal of Approximate Reasoning 31, 291–311 (2002)
6. de Campos, L.M., Puerta, J.M.: Stochastic local search algorithms for learning belief networks: Searching in the space of orderings. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, pp. 228–239. Springer, Heidelberg (2001)
7. de Campos, L.M., Puerta, J.M.: Local Search Methods for Learning Bayesian Networks Using a Modified Neighborhood in the Space of DAGs. In: Garijo, F.J., Riquelme, J.-C., Toro, M. (eds.) IBERAMIA 2002. LNCS (LNAI), vol. 2527, pp. 182–192. Springer, Heidelberg (2002)
8. Chickering, D.M.: Optimal structure identification with greedy search. Journal of Machine Learning Research 3, 507–554 (2002)
9. Chickering, D.M., Geiger, D., Heckerman, D.: Learning Bayesian networks is NP-Complete. In: Fisher, D., Lenz, H. (eds.) Learning from Data: Artificial Intelligence and Statistics V, pp. 121–130. Springer, Heidelberg (1996)
10. van Dijk, S., van der Gaag, L.C., Thierens, D.: A Skeleton-Based Approach to Learning Bayesian Networks from Data. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 132–143. Springer, Heidelberg (2003)
11. Consortium, E.: Elvira: An Environment for Creating and Using Probabilistic Graphical Models. In: Proc. of the First European Workshop on Probabilistic Graphical Models, pp. 222–230 (2002)
12. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning 29, 131–163 (1997)
13. Friedman, N., Nachman, I., Peer, D.: Learning Bayesian networks from massive datasets: The "sparse candidate" algorithm. In: Proc. of the 15th Conf. on Uncertainty in Artificial Intelligence, pp. 201–210 (1999)
14. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning 20, 197–244 (1995)
15. Gámez, J.A., Puerta, J.M.: Constrained Score+(Local)Search Methods for Learning Bayesian Networks. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 161–173. Springer, Heidelberg (2005)
16. Jensen, F.V.: Bayesian Networks and Decision Graphs. Springer, Heidelberg (2001)
17. Neapolitan, R.: Learning Bayesian Networks. Prentice-Hall, Englewood Cliffs (2003)

18. Margaritis, D.: Learning Bayesian Model Structure from Data. Ph.D. thesis, School of Computer Science, Carnegie Mellon University (2003)
19. Moral, S.: An Empirical Comparison of Score Measures for Independence. In: Proc of the 10th IPMU International Conference, pp. 1307–1314 (2004)
20. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo (1988)
21. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. Lecture Notes in Statistics, vol. 81. Springer, Heidelberg (1993)
22. Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max-min hill-climbing Bayesian network structure learning algorithm. Machine Learning 65(1), 31–78 (2006)

On Directed and Undirected Propagation Algorithms for Bayesian Networks

Christophe Gonzales[1], Khaled Mellouli[2], and Olfa Mourali[3]

[1] LIP6 – université Paris 6 – France
[2] IHEC – Carthage – Tunisie
[3] ISIG – Université de Kairouan

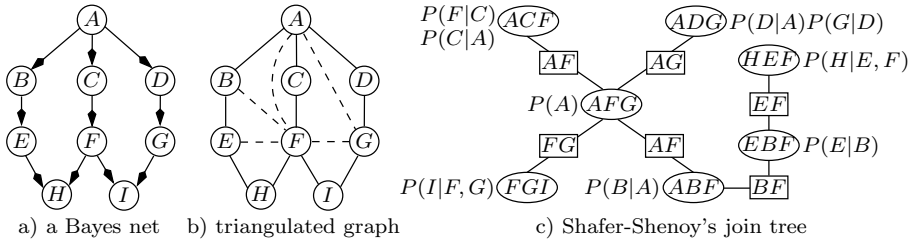
Abstract. Message-passing inference algorithms for Bayes nets can be broadly divided into two classes: i) *clustering* algorithms, like Lazy Propagation, Jensen's or Shafer-Shenoy's schemes, that work on secondary undirected trees; and ii) *conditioning* methods, like Pearl's, that use directly Bayes nets. It is commonly thought that algorithms of the former class always outperform those of the latter because Pearl's-like methods act as particular cases of clustering algorithms. In this paper, a new variant of Pearl's method based on a secondary directed graph is introduced, and it is shown that the computations performed by Shafer-Shenoy or Lazy propagation can be precisely reproduced by this new variant, thus proving that directed algorithms can be as efficient as undirected ones.

1 Introduction

In the last years, Bayesian nets (BN) [1,2,3] have become an increasingly popular knowledge representation framework for reasoning under uncertainty. They combine a directed acyclic graph (DAG) encoding a decomposition of a joint probability distribution over a set of random variables with powerful exact inference techniques [3,4,5,6,7,8,9,10,11,12,13,14]. These can answer various queries including *belief updating*, i.e., computing the posterior probability of every variable given a set of observations, *finding the most probable explanation*, i.e., finding a maximum probability assignment of the unobserved variables, *finding the maximum a posteriori hypothesis*, i.e., finding an assignment to a subset of unobserved variables maximizing their probability. This paper will be restricted to belief updating.

Although the BN's graphical structure is very efficient in its ability to provide a compact storage of the joint probability, it is not well suited for probabilistic computations when the DAG is multiply-connected [15]. A much more efficient structure called a *join* or *junction tree* and representing an alternative decomposition of the joint probability has been introduced in the 90's [16,9] to serve as a support for inference algorithms [9,10,12,17] (Fig. 1.c). Unlike the original BN, this new structure is undirected and, since [15]'s paper, the idea that propagation based on undirected graphs always outperform the variants of Pearl's algorithm (based on directed graphs) [18,19,3,20] has often been conveyed in the literature. However, in a BN, the arc orientations provide some independence

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 598–610, 2007.



a) a Bayes net b) triangulated graph c) Shafer-Shenoy's join tree

Fig. 1. A Bayesian network and one of its join trees

information called *d-separation* that can be effectively exploited to reduce the inference computational burden [21] but that is lost by junction trees. For instance, in Fig. 1.a, if the value of A is known, then d-separation asserts that B is independent of C, D, F, G and I. Hence, upon receiving a new evidence on B, only the *a posteriori* probabilities of E and H need be updated.

The aim of this paper is to show how undirected inference techniques such as Shafer-Shenoy's method [12,17] or Lazy Propagation [10] can be viewed as a variant of Pearl's algorithm. More precisely, it is shown that the computations performed by both algorithms in a join tree derived from a variable elimination technique similar to [7] can be precisely reproduced by Pearl with local conditioning performed on a particular DAG. The advantage of translating undirected inference techniques into directed ones is that d-separation can easily be applied to speed-up computations (see above). As for Lazy Propagation, which already uses d-separation, the advantage lies in the possibility of improving online triangulations or even avoiding them. Moreover, keeping the secondary structure used for computations as close as possible to the original one is attractive as it minimizes the quantity of information lost passing from one structure to the other (triangulation actually loses d-separation informations). This can prove useful for instance in hybrid propagation methods [22,23] where approximate algorithms are used on some subgraphs of the BN, to select the most appropriate stochastic algorithm for each approximated subgraph, e.g., in some particular cases, it can be proved that logic sampling converges faster than Gibbs sampling.

The paper is organized as follows: Section 2 presents BN and describes Shafer-Shenoy's method. Section 3 illustrates on an example how the same computations can be conducted using a particular DAG and a general scheme for constructing this DAG is derived. Section 4 presents Pearl's method with local conditioning and shows why a new variant, when applied on such DAG, corresponds to Shafer-Shenoy. Section 5 extends these results to binary join trees and to Lazy Propagation. Finally Section 6 concludes the paper.

2 Bayesian Networks and Shafer-Shenoy's Algorithm

Definition 1. *A Bayesian network is a triple* $(\mathcal{V}, \mathcal{A}, \mathcal{P})$*, where* $\mathcal{V}=\{X_1, \ldots, X_n\}$ *is a set of random variables;* $\mathcal{A} \subseteq \mathcal{V} \times \mathcal{V}$ *is a set of arcs which, together with*

$\mathcal{V}$, constitutes a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$; $\mathcal{P} = \{P(X_i|\mathrm{Pa}_i) : X_i \in \mathcal{V}\}$ is the set of conditional probability matrices of each random variable X_i given the values of its parents Pa_i in graph $\mathcal{G}$. The BN represents a joint probability distribution over $\mathcal{V}$ having the product form: $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|\mathrm{Pa}_i)$.

Thus, in the BN of Fig. 1.a, $P(\mathcal{V})$ can be decomposed as $P(A)P(B|A)P(C|A)$ $P(D|A)P(E|B)P(F|C)P(G|D)P(H|E,F)P(I|F,G)$. Shafer-Shenoy's algorithm uses a secondary undirected structure called a join (or junction) tree to perform probabilistic inference, see Fig. 1.c. As shown by [24], this structure can always be constructed from a triangulated graph (Fig. 1.b) resulting from an elimination sequence of the random variables. Here, we used H,I,E,D,C,B,A,F,G. The cliques (resp. separators) of the join tree, i.e., the ellipses (resp. rectangles), are initially filled with functions (called *potentials*) of the variables they contain. Usually, cliques are filled with the conditional probabilities of the BN and separators with unity tables, i.e., tables filled with 1's. Shafer-Shenoy then performs inference by sending messages in both directions along the edges of the junction tree. A message sent from a clique C_i to an adjacent clique C_j through separator $S_{ij} = C_i \cap C_j$ is computed by multiplying the potentials stored in C_i by the messages received from all the adjacent cliques of C_i except C_j and then summing out the result over the variables not in S_{ij}. For instance, on Fig 2, message ❶ corresponds to $\sum_G P(A) \times ② \times ④ \times ⑤$. Semantically, these operations correspond to marginalizing out from the joint probability the variables in $C_i \backslash S_{ij}$.

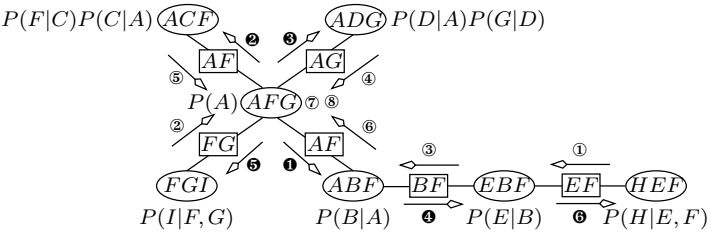


Fig. 2. The join tree and Shafer-Shenoy's inward and outward pass

As for the order in which messages are generated, Shafer-Shenoy advocates to use an asynchronous algorithm, but we prefer presenting here a collect/diffusion (or inward/outward) method as it is more convenient for the next Sections and it is well known that both schemes produce the same set of messages:

Algorithm 1 (generation of messages)

1. *choose an arbitrary clique, e.g., AFG, as the current clique;*
2. *inward pass: the current clique asks its adjacent cliques for their messages. In turn, they recursively ask their other adjacent cliques for messages. When a clique has received all the messages it waited for, it sends its own message.*
3. *outward pass: after the inward pass, clique AFG sends messages to its neighbors; they recursively send messages to their other adjacent nodes, and so on.*

Following the elimination sequence mentioned above, the messages sent during the inward and outward pass are computed as shown in the Table below. In this table, messages like $\mathbb{1}_{\mathcal{K}}$ represents $|\mathcal{K}|$-matrices filled with 1's and those like $P(\mathcal{T})_{\mathcal{K}}$ represent $|\mathcal{T}| \times |\mathcal{K}|$-matrices filled with $P(\mathcal{T})$ for every value of $\mathcal{K}$:

Table 1. Shafer-Shenoy's inward and outward pass computations

in:	elim node	sending clique	computation	message
	H	HEF	$\mathbb{1}_{EF} = \sum_H P(H\|E,F)$	①
	I	FGI	$\mathbb{1}_{FG} = \sum_I P(I\|F,G)$	②
	E	EBF	$\mathbb{1}_{BF} = \sum_E P(E\|B) \times$ ①	③
	D	ADG	$P(G\|A) = \sum_D P(D\|A)P(G\|D)$	④
	C	ACF	$P(F\|A) = \sum_C P(F\|C)P(C\|A)$	⑤
	B	ABF	$\mathbb{1}_{AF} = \sum_B P(B\|A) \times$ ③	⑥

out:	message	computation
	❶	$P(A,F) = \sum_G P(A)\mathbb{1}_{FG}P(G\|A)P(F\|A) = \sum_G P(A) \times$ ② $\times$ ④ $\times$ ⑤
	❷	$P(A)_F = \sum_G P(A)\mathbb{1}_{FG}P(G\|A)\mathbb{1}_{AF} = \sum_G P(A) \times$ ② $\times$ ④ $\times$ ⑥
	❸	$P(A)_G = \sum_F P(A)\mathbb{1}_{FG}P(F\|A)\mathbb{1}_{AF} = \sum_F P(A) \times$ ② $\times$ ⑤ $\times$ ⑥
	❹	$P(B,F) = \sum_A P(B\|A)P(A,F) = \sum_A P(B\|A) \times$ ❶
	❺	$P(F,G) = \sum_A P(A)P(G\|A)P(F\|A)\mathbb{1}_{AF} = \sum_A P(A) \times$ ④ $\times$ ⑤ $\times$ ⑥
	❻	$P(E,F) = \sum_B P(E\|B)P(B,F) = \sum_B P(E\|B) \times$ ❹

Computation of *a posteriori* marginal probabilities is performed in a similar way, except that new informations (evidence) about some random variables are entered into cliques as if they were part of the product decomposition of $P(\mathcal{V})$.

3 Constructing a New DAG for Shafer-Shenoy

The aim of this Section is to provide a generic algorithm based on a DAG that produces precisely the same computations as those of the preceding Section. Note that, for the moment, this algorithm is not related to Pearl's method. This will be the topic of the next Section. Here, our purpose is only the construction of a new graph. This one is usually different from the original BN and, to explain how it can be derived from the latter, it is best mimicking Shafer-Shenoy's algorithm using the same elimination ordering as before.

Before any elimination occurs, the conditional probabilities of the product decomposition of $P(\mathcal{V})$ are stored in the nodes of the BN as shown in Fig. 3.a. As mentioned in the preceding Section, eliminating variable H (resp. I) amounts to substitute $P(H|E,F)$ (resp. $P(I|F,G)$) by $\sum_H P(H|E,F) = \mathbb{1}_{EF}$ (resp. $\sum_I P(I|F,G) = \mathbb{1}_{FG}$). Such operations can be performed on the BN by replacing the probabilities stored in H and I by $\mathbb{1}_{EF}$ and $\mathbb{1}_{FG}$ (Fig. 3.b). As shown in Table 1, eliminating E is achieved by computing $\mathbb{1}_{BF} = \sum_E P(E|B)\mathbb{1}_{EF}$. If a node in the BN had the knowledge of both $P(E|B)$ and $\mathbb{1}_{EF}$, it would be able to perform this operation. Unfortunately, $\mathbb{1}_{EF}$ and $P(E|B)$ are stored in nodes H and E respectively. Hence either a message containing $P(E|B)$ should be sent

to H or a message containing $\mathbb{1}_{EF}$ should be sent to E. In this paper, to decide between these alternatives, the following rule will always be applied :

Rule 1. *Assume an algebraic operation F on some matrices stored into nodes $X_{i_1}, \ldots, X_{i_k}$ of $\mathcal{V}$ needs to be performed. Let X_{i_p} be any node such that no X_{i_q}, $q \neq p$, is a descendant of X_{i_p}, i.e., X_{i_q} cannot be reached from X_{i_p} following a sequence of arcs (along their directions). Then all the X_{i_q}'s, $q \neq p$, will send to X_{i_p} a message containing the matrix they store, and X_{i_p} will perform F.*

Using rule 1, to mimic the elimination of node H, node E must send to H message $P(E|B)$ and H computes $\mathbb{1}_{BF} = \sum_E P(E|B)\mathbb{1}_{EF}$. H then replaces its own probability $\mathbb{1}_{EF}$ by $\mathbb{1}_{BF}$. As E sent its conditional probability, it need not store anything anymore. Similarly, when eliminating D, node D should send message $P(D|A)$ and G should substitute $P(G|D)$ by $P(G|A) = \sum_D P(D|A)P(G|D)$. The elimination of C leads to C sending message $P(C|A)$ to F and F replacing $P(F|C)$ by $P(F|A) = \sum_C P(F|C)P(C|A)$. Of course, neither C nor D should store a conditional probability anymore since they already transmitted their own (Fig. 3.c). The elimination of B should be performed by computing $\mathbb{1}_{AF} = \sum_B P(B|A)\mathbb{1}_{BF}$. As $\mathbb{1}_{BF}$ and $P(B|A)$ are stored in H and B respectively, by rule 1, B should send a message to H. This implies adding a new arc (B, H) as illustrated on Fig. 3.d. Moreover, as after sending its message to H, B does not store a matrix anymore, it will never send any other message, hence arc (B, E) can be safely removed. This is illustrated by representing (B, E) by a dashed arc. Eliminating A requires several messages sent to either H or G (here rule 1 cannot settle), H was chosen arbitrarily on Fig. 3.e. Finally, eliminating F can be performed either by H transmitting a message to I or the converse. This example justifies the following scheme for constructing the new DAG:

Algorithm 2 (construction of a directed secondary structure)
1. *Assign to every node X_k of the original BN a label $L(X_k) = \{X_k\} \cup \mathrm{Pa}_k$;*
2. *For every node X_k, in their order of elimination, let $\mathcal{V}_{X_k} = \{X_{k_1}, \ldots, X_{k_p}\}$ be the set of nodes the labels of which contain X_k. Select among $\mathcal{V}_{X_k}$ a node X_i according to rule 1 and, for every node X_j in $\mathcal{V}_{X_k} \backslash \{X_i\}$, add an arc (X_j, X_i) if necessary, remove the other arcs outcoming from X_j. Let $L(X_i) = \cup_{X_{k_j} \in \mathcal{V}_{X_k}} L(X_{k_j}) \backslash \{X_k\}$ and let $L(X_{k_j}) = \emptyset$ for all $k_j \neq i$.*

The following lemma summarizes this Section:

Lemma 1. *Shafer-Shenoy's inward pass can be precisely reproduced by constructing the DAG resulting from Algorithm 2, sending messages downward along the solid arcs of this DAG and computing new messages as described above, the latter being obtained by multiplying the messages received by a node by the conditional probability stored into the node.*

4 A New Variant of Pearl's Algorithm

Pearl's-like methods are applied on DAG such as a BN but, as they need singly-connected networks, i.e., graphs without loops, to produce correct answers, they

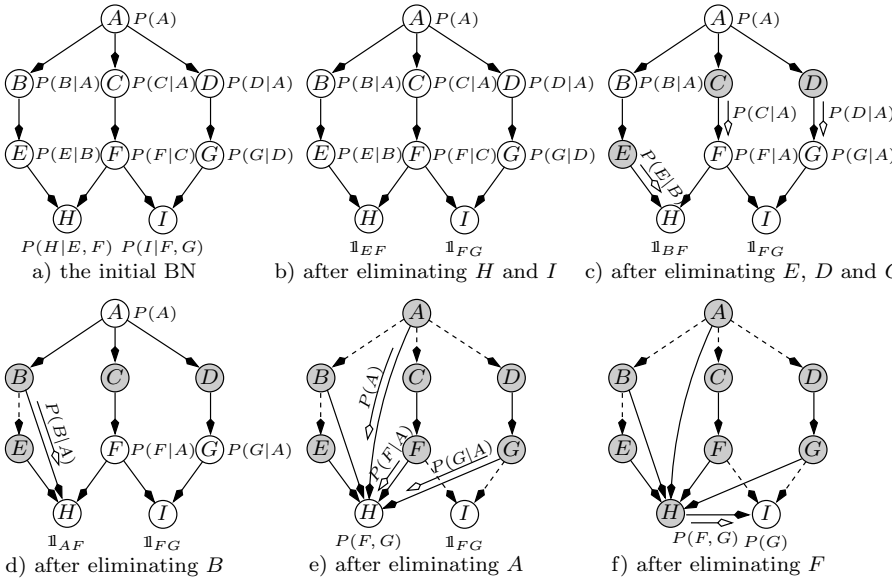


Fig. 3. The construction of a directed secondary structure

often use a preprocess called *cutset conditioning* that transforms the BN into a singly-connected graph on which computations are then performed. The transformation advocated by [19] consists of applying the following algorithm:

Algorithm 3 (local cutset). *Let $(\mathcal{V}, \mathcal{A}, \mathcal{P})$ be a BN. Select some arcs in graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ the removal of which keeps the graph connected while removing all cycles. Assign to every remaining arc (X_i, X_j) initial label X_i. For each arc (X_k, X_j) removed, there still exists exactly one trail joining X_k to X_j, i.e., a sequence of arcs that, without taking into account their directions, can be followed to reach X_j from X_k. Add X_k to the label of every arc on this trail.*

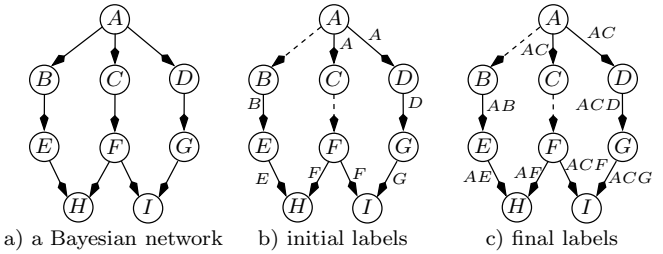
For instance, applying Algorithm 3 on the graph of Fig. 4.a results in the graph of Fig. 4.c: arcs (A, B) and (C, F) have been chosen arbitrarily to be removed. Fig. 4.b shows the initial labels of the remaining arcs. Trail B, E, H, F, I, G, D, A joins B and A, hence A is added to the label of every arc of this trail. Similarly, trail F, I, G, D, A, C joins F and C, thus C should be added to the label of every arc of this trail, hence resulting in Fig. 4.c. Once labels have been established, [19] advocates to use the following propagation algorithm:

Algorithm 4 (Pearl's-like method with local conditioning)

1. *Select an arbitrary node, say X_i, in the graph resulting from Algorithm 3 (for instance node I in the graph of Fig. 4.c).*
2. *inward pass: the current node asks its adjacent nodes for their messages. In turn, they recursively ask their other adjacent nodes for messages. When a node has received all the messages it waited for, it sends its own message.*

3. outward pass: after the inward pass, node X_i sends to its neighbors messages; they recursively send messages to their other adjacent nodes, and so on.

A message sent by a node X_j to one of its children (resp. parents) X_k is the sum over the variables not belonging to the label of arc (X_j, X_k) (resp. (X_k, X_j)) of the product of $P(X_j|\mathrm{Pa}_j)$ by all the messages sent to X_j except that sent by X_k.



a) a Bayesian network b) initial labels c) final labels

Fig. 4. Local conditioning and arc labeling

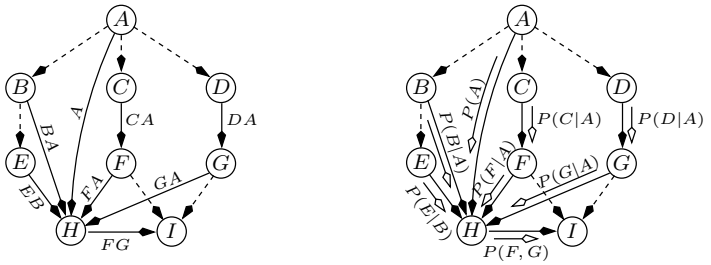
For instance, in Fig. 4.c, I would send a message to F equal to $\lambda(ACF) = \sum_{IG} P(I|F,G)\pi(ACG)$, where $\pi(ACG)$ is the message sent to I by G. Note that $\lambda(ACF)$ and $\pi(ACG)$ are messages of size $A \times C \times F$ and $A \times C \times G$ respectively. Thus arc labels indicate the size of messages sent throughout the network.

Now, let us come back to the unification of Pearl's and Shafer-Shenoy's algorithms. Applying the labeling algorithm below, which is a simple variant of Algorithm 3, to the secondary structure resulting from Algorithm 2, we obtain the graph of Fig. 5.a. It is striking that the labels correspond precisely to the size of the messages sent by Shafer-Shenoy, as described in the preceding Section.

Algorithm 5 (secondary structure labeling). *Let $\mathcal{G}$ be a BN and $\mathcal{G}'$ be the result of the application of Algorithm 2 to $\mathcal{G}$. For every arc (X_i, X_j) in $\mathcal{G}'$, assign label $\{X_i\}$ if it also belongs to $\mathcal{G}$, else $\emptyset$. For each arc (X_k, X_j) removed, add X_k to the label of every arc on the trail still joining X_j to X_k.*

The messages of the inward pass of Algorithm 4 performed on the graph of Fig. 5.a are precisely the same as those of Shafer-Shenoy. For instance, the message from H to I is equal to $\sum_{A,B,E,H} P(E|B) \times P(B|A) \times P(A) \times P(F|A) \times P(G|A) \times P(H|E,F) = \sum_A P(A)P(F|A)P(G|A) \times \sum_B P(B|A) \times \sum_E P(E|B) \times \sum_H P(H|E,F)$. The last sums are those of Shafer-Shenoy as they correspond to the messages and computations mentioned in Fig. 3. This suggests that Pearl can perform the same computations as Shafer-Shenoy when appropriately ordering its sequence of products and summations. This is, in essence, quite similar to the algorithm in [25], except that we use BN for inference rather than a secondary structure related only to computations and not to the original graph.

For the outward pass, Algorithm 4 will also produce messages of the same size as Shafer-Shenoy. However, if care is not taken, computations may be more time-consuming than in Shafer-Shenoy. For instance, assuming that I is selected



a) labels of the secondary structure e) Shafer-Shenoy's messages

Fig. 5. Arc labeling on the new Shafer-Shenoy's DAG

at step 1, the outward pass will start by I sending to H message $\sum_I P(I|F,G) = \mathbb{1}_{FG}$. The message sent by H to G is:

$$\pi_{F,G} = \sum_{A,B,E,F,H} P(E|B)P(B|A)P(A)P(H|E,F)\mathbb{1}_{FG}P(F|A) \quad (1)$$
$$= \sum_{A,F} P(A)P(F|A)\mathbb{1}_{FG} \sum_B P(B|A) \sum_E P(E|B) \sum_H P(H|E,F) \quad (2)$$
$$= \sum_{A,F} P(A)P(F|A)\mathbb{1}_{FG}\mathbb{1}_{AF}, \quad (3)$$

the latter being message ❶. Now, to actually perform the computations of (3) rather than those of (1), remind the sequence of messages sent to H, i.e., first $P(E|B)$ from E, then $P(B|A)$ from B, and finally $P(A)$, $P(F|A)$ and $P(G|A)$ from A, F and G. The corresponding products/sums performed when these messages were received are described in Table 2. Remark that (3) corresponds to multiplying $P(A)P(F|A)$ by the output of the penultimate computation (the third one) and by the message sent by I, and then to summing out unwanted variables. When H computes messages sent to F and A, the same element of the stack $\mathbb{1}_{AF}$ can be used to calculate $\sum_{A,G} P(A)P(G|A)\mathbb{1}_{FG}\mathbb{1}_{AF}$ and $\sum_{F,G} \mathbb{1}_{FG}\mathbb{1}_{AF}P(F|A)P(G|A)$. The former message corresponds to Shafer-Shenoy's ❷. The latter is never sent by Shafer-Shenoy, but it will eventually be computed to obtain the marginal probability of A, F or G as it corresponds to the product of the messages sent to clique AFG or, when multiplied by $P(A)$, to the product of the messages sent via separators AF, FG and AF. Similarly, when H computes the message sent to B, it should use the second element of the stack to avoid computing twice $\sum_H P(H|E,F)$. Thus, to behave as Shafer-Shenoy, Pearl should store in each node a stack of the temporary computations performed during the inward pass (see Table 2) and use this stack during the outward pass. Note that this does not actually require more space than in Shafer-Shenoy's algorithm since, in the latter, these temporary matrices are stored within separators.

Additional savings can be gained observing that once H has sent messages to A, F, G, the product of the messages it received from I and these nodes will be used for computing those sent to E and B. Thus, provided each time a node sends messages to some other nodes it keeps track of the product of the

Table 2. H's stack of temporary inward pass computations

stack index	stack content	sender
1	$\mathbb{1}_{EF} = \sum_H P(H\|E,F)$	
2	$\mathbb{1}_{BF} = \sum_E P(E\|B)\mathbb{1}_{EF}$	E
3	$\mathbb{1}_{AF} = \sum_B P(B\|A)\mathbb{1}_{BF}$	B
4	$P(F,G) = \sum_A P(F\|A)P(G\|A)P(A)\mathbb{1}_{AF}$	A,F,G

messages the latter sent it, it can be shown that Pearl will perform the same computations as Shafer-Shenoy. For instance, once messages to A, F and G have been sent, node H can store $P(A,F) = \sum_G P(A)P(F|A)P(G|A)\mathbb{1}_{FG}$. Then it can compute $\sum_{A,F} P(A,F)\mathbb{1}_{BF}$, which is precisely the message it should send to B. Note that it also corresponds to the product of the messages sent to clique ABF by Shafer-Shenoy and that $\mathbb{1}_{BF}$ is the top of H's stack when matrix $\mathbb{1}_{AF}$ has been popped. Then H can store $P(B,F) = \sum_A P(B|A)P(A,F)$, and it can send to E message $\sum_F P(B,F)\mathbb{1}_{EF}$, which corresponds to the product of the messages sent to clique EBF by Shafer-Shenoy. Note again that $\mathbb{1}_{EF}$ is the top of H's stack once $\mathbb{1}_{BF}$ has been popped. Finally, messages sent to C and D correspond to marginalizations of the messages sent to cliques ADG and ACF. This justifies the following Proposition:

Proposition 1 (unification of Pearl and Shafer-Shenoy). *Let $\mathcal{G}$ be a BN. Let $\mathcal{G}'$ be the result of the application of Algorithms 2 and 5 on $\mathcal{G}$, according to an elimination ordering σ. Assign to each node X_i an empty stack $\mathcal{T}(X_i)$. Using the two passes below, Pearl performs the same computations as Shafer-Shenoy:*

Inward pass: *Messages are like in Algorithm 4. For each message sent by a node X_i, let $\mathcal{Q}$ be the union of $P(X_i|\mathrm{Pa}_i)$ and the set of messages received by X_i. Let S be the set of variables of these factors ordered according to σ. For every $X_k \in S$, select the factors in $\mathcal{Q}$ that contain X_k; remove them from $\mathcal{Q}$, multiply them and sum the result over X_k. Add the result to $\mathcal{Q}$ and to X_i's stack, and indicate which senders sent these factors.*

Outward pass: *Messages are like in Algorithm 4. For each node X_i in σ's reverse order, let M be the message received by X_i during the outward pass (if any). Compute messages to X_i's adjacent nodes as follows: Let S be the variables in the "sender" column at the top of X_i's stack $\mathcal{T}(X_i)$. For all nodes X_k in S, send a message to X_k equal to the sum over the variables not in the label of arc (X_k, X_i) of the product of M by the messages sent to X_i by nodes in $S\backslash\{X_k\}$ and the stack content of the element just under the top if it exists else $P(X_i|\mathrm{Pa}_i)$. After messages have been sent to all nodes in S, pop X_i's stack once. If the sender's column is empty, pop it again. Update M by multiplying it by the messages sent to X_i by all nodes in S and sum over the variables not belonging to any factor in $\mathcal{T}(X_i)$. Iterate the process until the stack is empty.*

5 Extension to Binary Join Trees and Lazy Propagation

It is well known that, in general, Shafer-Shenoy's algorithm is slower than Jensen's and that, to be competitive, it needs to be run in a binary join tree, that is, in a tree where no node has more than 3 neighbors. Algorithms do exist to map a general join tree into a binary one [17]. The problem with general join trees comes from the outward phase and is illustrated on Fig. 6.a: assume that messages ①, ②, ③ and ④ were sent during the inward phase, then messages ❶, ❷, ❸ and ❹ of the outward phase are computed as:

$$❶ = \sum_F P(A) \times ① \times ② \times ③ \qquad ❷ = \sum_G P(A) \times ① \times ② \times ④,$$
$$❸ = \sum_A P(A) \times ① \times ③ \times ④ \qquad ❹ = \sum_G P(A) \times ② \times ③ \times ④.$$

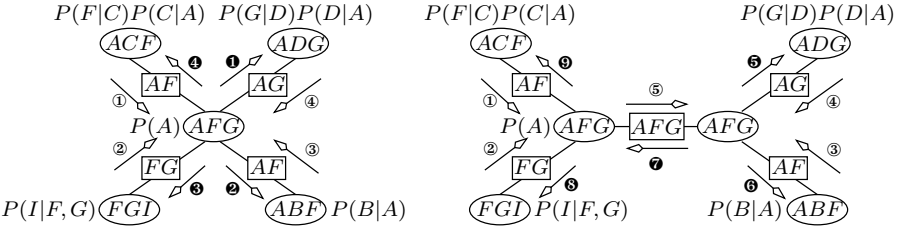


Fig. 6. Join trees vs. binary join trees

Remark that some products are done several times. To avoid these redundancies, Shenoy proposes to modify the join tree so that no node has more than 3 neighbors (Fig. 6.b). Computations in this new structure are then:

$$⑤ = P(A) \times ① \times ② \qquad ❺ = \sum_F ③ \times ⑤ \qquad ❻ = \sum_G ④ \times ⑤,$$
$$❼ = ③ \times ④ \qquad ❽ = \sum_A P(A) \times ❼ \times ① \qquad ❾ = \sum_G P(A) \times ❼ \times ④.$$

Actually, message ❼ captures the idea that product ③ × ④ should not be performed several times. But avoiding these redundancies can also be obtained by observing that if we multiply during the inward phase the messages that arrive one by one and we store these results into a stack (see the left part of Table 3), then the outward phase messages can be computed by using this stack and multiplying by the messages on the right part of the Table. As is noticeable, from bottom up, these products can be computed incrementally requiring only one product for each line. This justifies the following proposition:

Proposition 2 (binary join tree unification). *Applying the algorithm of Proposition 1 with the rules below is equivalent to performing Shafer-Shenoy in a binary join tree:*

Inward pass: *When messages arrive to node X_i, perform the products one by one and store all of them into X_i's stack.*

Outward pass: *send outward messages to X_i's neighbor in the reverse order in which these neighbors sent their messages during the inward phase.*

Table 3. Binary join trees and the product of messages

inward sending clique	inward phase products	outward phase products
initial potential	$P(A)$	$\times ❷ \times ❸ \times ❹ = ❾$
ACF	$P(A) \times ①$	$\times ❸ \times ❹ = ❽$
FGI	$P(A) \times ① \times ②$	$\times ❹ = ❻$
ABF	$P(A) \times ① \times ② \times ③$	$= ❺$

The second extension we should mention is the unification with Lazy Propagation [10]. The latter is essentially similar to Jensen's or Shafer-Shenoy's algorithms in that it uses a join tree to perform propagation of potentials. However, it departs from these algorithms in the following manner:

1. Instead of storing only one potential in each clique, it stores a list of potentials (or conditional probability tables). It performs products on some potentials only when necessary, i.e., when they contain a variable that is to be marginalized-out.
2. It recognizes summations like $\sum_H P(H|T)$ the result of which is known for sure to be 1.
3. It uses *d*-separation to avoid sending a message from one clique to another if the random variables of these cliques are independent due to some evidence.

In our algorithm, using *d*-separation is quite obvious since we are working on a BN, and avoiding computing unity summations is easy: it only requires to know which variables of the conditional probability tables are on the left of conditioning bars. As for the first feature, our algorithm can be easily adapted: it is sufficient manipulate lists of potentials instead of performing directly all the products. Hence, Pearl's algorithm can be adapted to behave as Lazy Propagation.

6 Conclusion

This paper has shown that Pearl's-like methods could be adapted using a new secondary directed structure and an appropriate ordering to perform the same computations as Shafer-Shenoy or Lazy Propagation, thus proving that directed inference methods could be as efficient as undirected ones. The advantage of using directed secondary structures is twofold: first, it enables to perform quite simply *d*-separation analyses and, secondly, it enables to limit the amount of information lost during triangulation by keeping the secondary structure as close as possible of the original one. For instance, there are cases in which even if the BN contains cycles, Pearl's algorithm can compute correctly all marginal *a posteriori* probabilities without needing any conditioning, e.g., when the parent nodes of the cycle sinks are independent. In such cases, working on a directed structure is better than working on an undirected one since the latter requires an unnecessary triangulation.

References

1. Cowell, R., Dawid, A., Lauritzen, S., Spiegelhalter, D.: Probabilistic Networks and Expert Systems. In: Statistics for Engineering and Information Science, Springer, Heidelberg (1999)
2. Jensen, F.: An introduction to Bayesian Networks. Taylor and Francis (1996)
3. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)
4. Allen, D., Darwiche, A.: New advances in inference by recursive conditioning. In: Proceedings of UAI, pp. 2–10 (2003)
5. D'Ambrosio, B., Shachter, R., Del Favero, B.: Symbolic probabilistic inference in belief networks. In: Proceedings of AAAI, pp. 126–131 (1990)
6. Darwiche, A., Provan, G.: Query DAGs: A practical paradigm for implementing belief network inference. In: Proceedings of UAI, pp. 203–210 (1996)
7. Dechter, R.: Bucket elimination: A unifying framework for several probabilistic inference algorithms. In: Proceedings of UAI, pp. 211–219 (1996)
8. Huang, C., Darwiche, A.: Inference in belief networks: A procedural guide. International Journal of Approximate Reasoning 15(3), 225–263 (1996)
9. Jensen, F., Lauritzen, S., Olesen, K.: Bayesian updating in causal probabilistic networks by local computations. Computational Statistics Quarterly 4, 269–282 (1990)
10. Madsen, A., Jensen, F.: LAZY propagation: A junction tree inference algorithm based on lazy inference. Artificial Intelligence 113(1-2), 203–245 (1999)
11. Poole, D., Zhang, N.: Exploiting contextual independence in probabilistic inference. Journal of Artificial Intelligence Research 18, 263–313 (2003)
12. Shafer, G.: Probabilistic expert systems. Society for Industrial and Applied Mathematics (1996)
13. Sharma, R., Poole, D.: Efficient inference in large discrete domains. In: Proceedings of UAI, pp. 535–542 (2003)
14. Zhang, N., Poole, D.: Inter-causal independence and heterogeneous factorization. In: Proceedings of UAI, pp. 606–614 (1994)
15. Shachter, R., Andersen, S., Szolovits, P.: Global conditioning for probabilistic inference in belief networks. In: Proceedings of UAI (1994)
16. Lauritzen, S., Spiegelhalter, D.: Local computations with probabilities on graphical structures and their application to expert systems. The Journal of The Royal Statistical Society – Series B (Methodological) 50(2), 157–224 (1988)
17. Shenoy, P.: Binary join trees for computing marginals in the Shenoy-Shafer architecture. International Journal of Approximate Reasoning 17(1), 1–25 (1997)
18. Diez, F.: Local conditioning in Bayesian networks. Artificial Intelligence 87, 1–20 (1996)
19. Faÿ, A., Jaffray, J.Y.: A justification of local conditioning in Bayesian networks. International Journal of Approximate Reasoning 24(1), 59–81 (2000)
20. Peot, M., Shachter, R.: Fusion and propagation with multiple observations in belief networks. Artificial Intelligence 48, 299–318 (1991)
21. Geiger, G., Verma, T., Pearl, J.: Identifying independence in Bayesian networks. Networks 20, 507–534 (1990)
22. Dawid, A., Kjærulff, U., Lauritzen, S.: Hybrid propagation in junction trees. In: Bouchon-Meunier, B., Yager, R.R., Zadeh, L.A. (eds.) IPMU 1994. LNCS, vol. 945, pp. 87–97. Springer, Heidelberg (1995)

23. Kjærulff, U.: HUGS: Combining exact inference and Gibbs sampling in junction trees. In: Proceedings of UAI (1995)
24. Rose, D.: Triangulated graphs and the elimination process. Journal of Mathematical Analysis and Applications 32, 597–609 (1970)
25. Bloemeke, M., Valtorta, M.: A hybrid algorithm to compute marginal and joint beliefs in Bayesian networks and its complexity. In: Proceedings of UAI, pp. 16–23 (1998)

Lexicographic Refinements of Sugeno Integrals

Didier Dubois and Hélène Fargier

IRIT, 118 route de Narbonne
31062 Toulouse Cedex, France
{dubois, fargier}@irit.fr

Abstract. This paper deals with decision-making under uncertainty when the worth of acts is evaluated by means of Sugeno integral on a finite scale. One limitation of this approach is the coarse ranking of acts it produces. In order to refine this ordering, a mapping from the common qualitative utility and uncertainty scale to the reals is proposed, whereby Sugeno integral is changed into a Choquet integral. This work relies on a previous similar attempt at refining possibilistic preference functionals of the max-min into a so-called big-stepped expected utility, encoding a very refined qualitative double lexicographic ordering of acts.

1 Introduction

In the framework of decision under uncertainty, it has been pointed out that information about preference and uncertainty in decision problems cannot always be quantified in a simple way, but only qualitative evaluations can sometimes be attained. As a consequence, the topic of qualitative decision theory is a natural one to consider [1]. A trade-off between purely symbolic and purely numerical approaches to ranking decisions consists in using a single qualitative scale for assessing beliefs and utilities. In [2] two possibilistic qualitative criteria, an optimistic and a pessimistic one, whose definitions only require a common linearly ordered scale for utility and uncertainty have been proposed. Later, a wider family of decision criteria called *monotonic utilities* and that encompasses both possibilistic decision criteria, have been investigated [3]. It is based, mathematically speaking, on a Sugeno integral [11]. Unfortunately, monotonic utilities in general and possibilistic utilities in particular, may suffer from a lack of decisiveness power: the principle of Pareto efficiency is not respected: namely, when two actions have the same potential consequence in some given, likely state of the world, they may be equally preferred by qualitative decision criteria, although one may have much better consequences than the other, in the remaining states of the world.

Some authors tried to use the idea of lexicographic refinements with Sugeno integral. Murofushi [7] considered refining Sugeno integral with respect to a capacity by a vector of Sugeno integrals with respect to a sequence of capacities. This supposes a much richer information than the one available in decision making under uncertainty. Grabisch [5] proposed to use the fact that Sugeno integral is a median to refine it by cancelling the median terms when equal and comparing the values around the median via a leximin. Here we try to generalize

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 611–622, 2007.

the weighted extension of the leximin and the leximax to possibilistic integrals, proposed by Fargier and Sabbadin [4]. These authors have indeed shown that in the possibilistic case the order induced on decisions can be refined by expected utility, so that the strict consistency with the Pareto principle can be recovered. It can thus be asked if the same question can be solved for Sugeno integrals since prioritized minimum and maximum are special cases of fuzzy integrals.

The paper is structured as follows. In Section 2, we first give some background on qualitative preference functionals, and point out the "drowning effect" that generates a lack of decisiveness power. It is shown that the only situations where Sugeno integral satisfies the principle of efficiency are degenerate. Section 3 presents lexicographic refinements of Sugeno integrals. In particular, it shows that for any Sugeno integral with respect to a capacity, there exists a Choquet integral with respect to the same ordering of events that refines it. Another kind of refinement based on Moebius transforms is then considered. Proofs are omitted for the sake of brevity.

2 Limitations of Qualitative Preference Functionals

2.1 Definitions

Let S be a set of potential states of the world and X be a set of possible consequences. $\mathcal{F} = X^S$ denotes the set of potential acts, associating to each possible state s a consequence $f(s) \in X$. In this paper, S and X are supposed to be finite. Three particular sub classes are worthwhile noticing.

- *Constant acts*: each such act is identified with some $x \in X$, i.e., $\forall s \in S, x(s) = x$
- We will also often refer to the notion of *compound act* . For any set of states A, fAg is the act defined by: $fAg(s) = f(s)$ for all $s \in A$, and $fAg(s) = g(s)$ for all $s \in \overline{A}$.
- For any pair of consequences x and y in X, xAy will denote for short the act defined by: $xAy(s) = x$ for all $s \in A$, and $xAy(s) = y$ for all $s \in \overline{A}$. Such a compound act will be called a *binary act*.

Consider again the set $\mathcal{F}$ of acts. In our framework, we assume that it is possible to evaluate uncertainty and preferences by means of a finite totally ordered scale $(L, \leq)$ whose top and bottom elements are denoted $\top$ and $\bot$ respectively. We will also write α_i the elements of L, with $(\alpha_0 = \top > \alpha_1 > \ldots > \alpha_l = \bot)$.

The mapping from the set of consequences to L is a utility function $\mu : X \to L$. It is supposed that the top and bottom elements of L are in $\mu(X) = \{\mu(x), x \in X\}$. If not, just add an ideal consequence denoted $\top$ and a totally bad consequence denoted $\bot$ to X, that will be identified with the bounds of L.

Uncertainty is assumed to be captured by means of a set function $\gamma : 2^S \to L$ which is a monotonic measure (or a *capacity*), i.e. is such that: $\gamma(\emptyset) = \bot$, $\gamma(S) = \top$, $A \subseteq B \Rightarrow \gamma(A) \leq \gamma(B)$. This kind of set-function is very general and represents the minimal requirement for the representation of partial belief. When

numerical, this family includes probability measures and most other well-known representations of partial belief (including belief and plausibility functions, necessity and possibility measures...).

In the following, we assume without loss of generality that for any $A \subseteq S$, there exists x such that $\gamma(A) = \mu(x)$ (if there is no such x in the original consequence set, just add an element to X having this utility value). This is a classical assumption of existence of a certainty equivalent for each lottery.

The utility of an act f can then be defined as a Sugeno integral [11], a qualitative counterpart of weighted sum, where the sum is replaced by a *sup* (a *max* in the finite case) and the product by an *inf* (a *min* in the finite case):

$$S_{\gamma,\mu}(f) = \max_i \min(\alpha_i, \gamma(F_i)) \tag{1}$$

where $F_i = \{s, \mu(f(s)) \geq \alpha_i)\}$. Sugeno integral computes the median of its arguments. For a binary act xAy where $x \succeq y$, $S_{\gamma,\mu}(xAy)$ is the median value in the set $\{\mu(x), \mu(y), \gamma(A)\}$.

This Sugeno integral thus defines a weak order on $\mathcal{F}$ (i.e. complete and transitive relation) representing the preferences of the Decision Maker over acts:

$$f \succeq^{sug}_{\gamma,\mu} g \Leftrightarrow S_{\gamma,\mu}(f) \geq S_{\gamma,\mu}(g). \tag{2}$$

When there is no ambiguity about γ and μ, we simply use the notation $\succeq^{sug}$.

2.2 Limitations of Qualitative Preference Functionals

As said in the introduction, monotonic utilities suffer from a lack of decisiveness power and can even fail to satisfy the principle of efficiency of Pareto. This general principle says that, if f is as least as good as g on each state, and better than g on some non null state s, then f should be strictly preferred to g. We put a restriction on null events, which by definition do not play any role in the decision. Formally, let $\succeq$ be a preference relation on $\mathcal{F}$:

Null Events. An event A is said to be null[1] with respect to a preference relation $\succeq$ on acts iff $\forall f, g, h \in \mathcal{F}, fAh \succeq gAh$

Remark that if A and B are null, so is $A \cup B$ and reciprocally. So, it can be said that a state s is null iff $\{s\}$ is null.

Weak Pareto Dominance. Act f weakly Pareto-dominates g (denoted $f \succeq_{Pareto} g$) iff $\forall s$ not null , $f(s) \geq g(s)$.

[1] If $\succeq$ is defined by a Sugeno integral, A null implies $\gamma(A) = \bot$ but $\gamma(A) = \bot$ does not imply that A is null. For instance let $S = \{s_1, s_2, s_3\}$ and let γ be the necessity measure built on the possibility distribution $\pi(s_i) = \top, \forall s_i$. Consider the acts $h = f = \top\{s_1\}\top$ and $g = \bot\{s_1\}\top$. Obviously, $\gamma(\{s_1\}) = \bot$. But s_1 is not a null state; indeed, $S_{\gamma,\mu}(f\{s_1\}h) = S_{\gamma,\mu}(f) = \top > S_{\gamma,\mu}(g\{s_1\}h) = S_{\gamma,\mu}(g) = \bot$. In the context of Sugeno integral, A null rather means that $\gamma(A \cup B) = \gamma(B)\ \forall B$.

Pareto Strict Preference. f dominates g according to Pareto ($f \succ_{Pareto} g$) iff $f \succeq_{Pareto} g$ and not $g \succeq_{Pareto} f$

The principle of efficiency of Pareto is then classically defined as an agreement with Pareto's strict preference.

Pareto Efficiency Principle. A preference relation $\succeq$ on $\mathcal{F}$ satisfies the principle of Pareto efficiency iff $f \succ_{Pareto} g \implies f \succ g$

Let us go back to the weak order induced by Sugeno integral. It is obviously in agreement with weak dominance, but not with strict dominance. Consider for instance an event A which is not certain but quite likely ($\gamma(A) > \bot$) and consequence x such $\mu(x) = \gamma(A) > \bot$. Then consider acts f, g that both ensure x on A but differ on the opposite event

$f : f(s) = x$ if $s \in A$, $f(s) = \bot$ if $s \notin A$, $\quad g : g(s) = x$ whatever $s \in S$.

Then $S_{\gamma,\mu}(g) = \mu(x)$, but also $S_{\gamma,\mu}(f) = \max(\min(\mu(x), \gamma(A)), \min(\mu(\bot), \gamma(\overline{S})))$ $= \mu(x)$. Hence $f \sim g$. This means that the fact that $\mu(g(s)) > \mu(f(s))$ on $\overline{A}$ is not taken into account. Technically, the weak utility of $\bot$ is "drowned" by the coefficient $\min(\mu(x), \gamma(A))$ in the computation of $S_{\gamma,\mu}(f)$. More generally, the standard expression of Sugeno integral (1) uses two operators that are monotonic but not strictly (namely, max and min), hence two nested drowning effects.

The drowning effect is also often understood as an incapacity to obey the well-known *Sure-Thing Principle* (STP) [9].

STP: $\forall f, g, h, h', fAh \succeq gAh \Leftrightarrow fAh' \succeq gAh'$

This principle indeed insures that identical consequences do not influence the relative preference between two acts, hence the impossibility of a drowning effect. The Pareto efficiency and Sure Thing principles are close to each other. Indeed, when the preference is complete and transitive (it is a weak order), as it is the case here, the STP is a sufficient condition for Pareto-efficiency.

It has been shown by Marichal [6] that the STP is generally not compatible with Sugeno integrals. We can moreover prove that the Sugeno integral is almost incompatible not only with the STP, but also with the less demanding principle of Pareto efficiency. This is the first result of this paper.

Theorem 1. *Under the assumption of existence of certainty equivalents for binary acts, then:* $\succeq^{sug}_{\gamma,\mu}$ *is Pareto-efficient if and only if there exist a unique state* s^* *such that* $\forall A, \gamma(A) = \top$ *if* $s^* \in A, \gamma(A) = \bot$ *if* $s^* \notin A$.

This means that Sugeno integral cannot be efficient unless applied when there is no uncertainty at all. These impossibility results are not necessarily damning. It is now established [4] that two of the Sugeno integrals, namely those that are defined upon a possibility distribution can be *refined* by an expected utility. This means that, when γ is a possibility measure, there always exist a probability distribution p and a (risk-prone) utility function u^* such that $f \succ^{sug}_{\gamma,\mu} g$ implies $f \succ_{EU^+} g$, EU^+ being the expected utility based on p and u^*. This expected utility confirms all strict preferences decided by the qualitative rule, but can break ties left by the latter. As any expected-utility-based preference relation,

it satisfies the Sure Thing Principle and thus the principle of Pareto efficiency. An alternative (risk-averse) utility function u_* and an expected utility EU^- can similarly be built if γ is a necessity measure. It can thus be asked if the same question can be solved for discrete Sugeno integrals since prioritized minimum and maximum are special cases of fuzzy integrals.

At a first glance, the answer is negative. One basic reason why prioritized maximin and minimax aggregations can be refined by a weighted average with fixed weights is that these operations do not violate the STP in a drastic way. Indeed the ordering relations induced by possibility and necessity measures satisfy a weaker independence condition:

$$\text{Axiom WSTP:} \forall f, g, h, h', fAh \succ gAh \Rightarrow fAh' \succeq gAh'.$$

So modifying two acts by altering their common consequences never results in a strong preference reversal. On the contrary, such a preference reversal is clearly possible for Sugeno integrals because for a fuzzy measure γ and three sets A, B, C, where C is disjoint from both A and B, one may have $\gamma(A) > \gamma(B)$ and $\gamma(B \cup C) > \gamma(A \cup C)$. This feature makes it impossible to refine rankings of acts induced by Sugeno integrals by means of another functional which satisfies the Sure Thing Principle. In particular, a Sugeno integral with respect to a given fuzzy measure cannot be refined by some expected utility with respect to a single probability distribution. Several lines can nevertheless be explored in order to partially recover efficiency, as shown in the next section.

3 Toward Lexicographic Refinements of the Sugeno Integral

The general idea is to define refinements of $\succeq^{sug}_{\gamma,\mu}$, i.e. relations $\succeq$ such that: $f \succ^{sug}_{\gamma,\mu} g \implies f \succ g$. For the reason laid bare just before, if $\succeq^{sug}_{\gamma,\mu}$ violates the WSTP, none of its refinements can satisfy it. We can nevertheless try to satisfy the so-called *Comonotonic Sure Thing Principle*. Recall that two acts f, g are comonotonic iff there exists a single permutation σ on the states of S that rearrange the values of both $\mu(f)$ and $\mu(g)$ in non-decreasing order, i.e. such that:

$$\mu(f(s_{\sigma(1)})) \leq \mu(f(s_{\sigma(2)})) \leq \cdots \leq \mu(f(s_{\sigma(n)}))$$

$$\mu(g(s_{\sigma(1)})) \leq \mu(g(s_{\sigma(2)})) \leq \cdots \leq \mu(g(s_{\sigma(n)}))$$

Hence the Comonotonic Sure Thing Principle stipulates:

$$\text{CSTP } \forall f, g, h, h' \text{ comonotonic }, \forall A \subseteq S : fAh \succeq gAh \iff fAh' \succeq gAh'$$

Sugeno integral generally does not satisfy CSTP, but it obeys its weak form forbidding preference reversals.

$$\text{WCSTP } \forall f, g, h, h' \text{ comonotonic }, \forall A \subseteq S : fAh \succ gAh \iff fAh' \succeq gAh'$$

It should thus be possible to get refinements that satisfy CSTP. The first idea consists in going back to the basics of Fargier and Sabbadin's work, and the so-called big-stepped transformation, that can be applied to any max-min form. The second idea is to exploit the similarity between expressions of Sugeno integral and of discrete Choquet integral [10][8]. In particular, while Choquet integrals are additive for comonotonic acts, Sugeno integrals are both maxitive and minitive for such acts. The natural idea is thus to look for a Choquet integral as a refinement of Sugeno integrals. Finally, the qualitative Moebius transform of a fuzzy measure can be turned into a probabilistic mass function via a transformation that directly yields a Choquet integral.

3.1 Lexicographic Refinements of Maximin Aggregations

Let $\overrightarrow{a} = (a_1, ..., a_n)$ and $\overrightarrow{b} = (b_1, ..., b_n)$ be some vector of evaluations using a common and finite ordered scale $L = (\alpha_0 = \top > \alpha_1 > \ldots > \alpha_l = \bot)$.

A usual way of escaping the drowning effect is to refine the ordering on vectors induced by the max (resp. min) aggregation using the leximax (resp. leximin) ordering. Let $\overrightarrow{a}, \overrightarrow{b} \in L^n$. Then

$$\overrightarrow{a} \succeq_{lmax} \overrightarrow{a} \Leftrightarrow \text{ or } \begin{cases} \forall j, a_{(j)} = b_{(j)} \\ \exists i, \forall j < i, a_{(j)} = b_{(j)} \text{ and } a_{(i)} > b_{(i)} \end{cases} \tag{3}$$

$$\overrightarrow{a} \succeq_{lmin} \overrightarrow{b} \Leftrightarrow \text{ or } \begin{cases} \forall j, a_{(j)} = b_{(j)} \\ \exists i, \forall j > i, a_{(j)} = b_{(j)} \text{ and } a_{(i)} > b_{(i)} \end{cases} \tag{4}$$

where, for any $\overrightarrow{w} \in L^n$, $w_{(k)}$ is the k-th *greatest* element of $\overrightarrow{w}$ (i.e. $w_{(1)} \geq \ldots \geq w_{(n)}$). In practice, applying a leximin (resp. leximax) comparison on vectors comes down to rearranging their components in increasing (resp. decreasing) order, then comparing the ordered vector lexicographically, hence the name leximin (resp. leximax).

Following [4] the same type of approach can be used to refine the ranking induced by a max-min aggregation of entries in matrices. Let us consider any weak order (i.e. complete and transitive relation) $\trianglerighteq$ on vectors of L^m. The definition of leximin and leximax procedures can be applied to matrices $n \times m$, since the rows of the matrices can be rearranged in increasing and decreasing order according to $\trianglerighteq$. So, denoting $\rhd$ (resp. $\equiv$) the strict (resp. symmetric) part of $\trianglerighteq$, we can compare any two matrices $[a]$ and $[b]$ according to $Leximin(\trianglerighteq)$ and $Leximax(\trianglerighteq)$:

$$[a] \succeq_{lmax(\trianglerighteq)} [b] \Leftrightarrow \text{ or } \begin{cases} \forall j, \overrightarrow{a}_{(j)} \equiv \overrightarrow{b}_{(j)} \\ \exists i, \forall j < i, \overrightarrow{a}_{(j)} \equiv \overrightarrow{b}_{(j)} \text{ and } \overrightarrow{a}_{(i)} \rhd \overrightarrow{b}_{(i)} \end{cases} \tag{5}$$

$$[a] \succeq_{lmin(\trianglerighteq)} [b] \Leftrightarrow \text{ or } \begin{cases} \forall j, \overrightarrow{a}_{(j)} \equiv \overrightarrow{b}_{(j)} \\ \exists i, \forall j > i, \overrightarrow{a}_{(j)} \equiv \overrightarrow{b}_{(j)} \text{ and } \overrightarrow{a}_{(i)} \rhd \overrightarrow{b}_{(i)} \end{cases} \tag{6}$$

where for any matrix $[w] \in (L^m)^n$, $\overrightarrow{w}_{(k)}$ is the k-th *greatest* line of $[w]$ according to $\trianglerighteq$ (i.e. $\overrightarrow{w}_{(1)} \trianglerighteq \ldots \trianglerighteq \overrightarrow{w}_{(n)}$). Now, just let $\trianglerighteq$ be the leximin or the leximax ranking of vectors (this is possible, since these relations are complete and transitive). Then, nested lexicographic ordering procedures $Leximax(\succeq_{lmin})$, $Leximin(\succeq_{lmax})$ can be recursively defined, in order to compare L-valued matrices.

Consider for instance the relation $\succeq_{lmax(\succeq_{lmin})}$ obtained by the procedure $Leximax(\succeq_{lmin})$. It applies to matrices of dimensions $n \times m$ with coefficients in $(L, \geq)$. In practice, the comparison comes down to rearranging $[a]$ and $[b]$ such that terms in each row are reordered increasingly w.r.t. $\geq$ and rows are arranged lexicographically top-down in decreasing order. Let $[a^\star]$ and $[b^\star]$ be rearranged matrices $[a]$ and $[b]$. Let $a^*_{i.}$ (resp. $b^*_{i.}$) be row i of a^* (resp. b^*). Then:

$$[a] \succeq_{lmax(\succeq_{lmin})} [b] \Leftrightarrow \text{ or } \begin{cases} \forall i, a^\star_{i.} =_{lmin} b^\star_{i.} \\ \exists k \leq p \text{ s.t. } \forall i < k, a^\star_{i.} =_{lmin} b^\star_{i.} \text{ and } a^\star_{k.} >_{lmin} b^\star_{k.} \end{cases}$$

Relation $\succeq_{lmax(\succeq_{lmin})}$ is a weak order. $[a] \simeq_{lmax(\succeq_{lmin})} [b]$ if and only if $[a*] = [b^*]$, i.e. both matrices have the same coefficients up to the above described rearrangement. As expected , $\succeq_{lmax(\succeq_{lmin})}$ refines the ranking obtained by the max-min aggregation:

$$\max_i \min_j a_{ij} > \max_i \min_j b_{ij} \text{ implies } [a] \succ_{lmax(\succeq_{lmin})} [b].$$

and especially, if $[a]$ Pareto-dominates $[b]$ in the strict sense ($\forall i, j, a_{ij} \geq b_{ij}$ and $\exists i^*, j^*$ such that $a_{i^*j^*} > b_{i^*j^*}$), then $[a] \succ_{lmax(\succeq_{lmin})} [b]$.

We can now show that this ordering can be encoded by a (double) big-stepped transformation, generalizing the result of [4] established for $m = 2$

Theorem 2. *There exist a transformation $\chi : L \to [0, +\infty)$ such that:*

$$[a] \succ_{lmax(\succeq_{lmin})} [b] \iff \sum_{i=1,n} \prod_{i=1,m} \chi(a_{i,j}) > \sum_{i=1,n} \prod_{i=1,m} \chi(b_{i,j})$$

As a matter of fact, the following transformation can be used to capture the leximax(leximin) ordering:

$$\chi^*(\alpha_l) = 0; \quad \chi^*(\alpha_i) = \frac{v}{N^{M^i+1}}, i = 0, k-1 \tag{7}$$

where $N \geq \max(2, n)$, $M \geq \max(2, m)$ and v is any positive normalization factor. For instance for $M = 2$ the series: $\frac{v}{N}$, $\frac{v}{N^2}$; $\frac{v}{N^4}$, $\frac{v}{N^8}$,... can be chosen. The choice of N, M, v does not really matter provided that M, N satisfy the constraints $N \geq \max(2, n)$ and $M \geq \max(2, m)$.

3.2 Capacity-Preserving Refinements of Sugeno Integrals

In the standard expression of Sugeno integral (equation (1)), the two operators max and min are monotonic but not strictly, hence two nested drowning

effects. The simplest idea to refine Sugeno integral is to use the $Leximax(\succeq_{lmin})$ procedure. This leads to the following decision rule:

$$f \succeq^{lsug} g \iff [f]_{\gamma,u} \succeq_{lmax(\succeq_{lmin})} [g]_{\gamma,u} \tag{8}$$

where $\forall f \in X^S, [f]_{\gamma,u}$ is a $(l+1) \times 2$ matrix on $L = (\alpha_0 > \alpha_1 > \ldots > \alpha_l)$ with coefficients $f_{i1} = \alpha_i$ and $f_{i2} = \gamma(F_{\alpha_i})$, $i = 0, l$.

The properties of $\succeq_{lmax(\succeq_{lmin})}$ are thus inherited:

Theorem 3. *$\succeq^{lsug}$ is a complete and transitive relation that refines the ranking of acts induced by $S_{\gamma,\mu}$. Moreover, $f \sim^{lsug} g$ iff $\forall \alpha, \gamma(F_\alpha) = \gamma(G_\alpha)$*

The last point indicates that acts equivalent with respect to stochastic dominance will not be discriminated by $\succeq^{lsug}$. Now, being a *Leximax($\succ_{lmin}$)* procedure, $\succeq^{lsug}$ can be encoded by a sum of products. We can for instance use the "big-stepping" function χ provided by equation 7. Let us set $N = l + 1$, $M = 2$ and choose the normalization factor so that $\chi(\gamma(S)) = 1$. We can now immediately derive a new evaluation function E^{lsug}, that provides a refinement of the ranking induced by $S_{\gamma,\mu}$:

$$E^{lsug}(f) = \sum_{\alpha \in L} \chi^*(\alpha) \cdot \chi^*(\gamma(F_\alpha)) \tag{9}$$

Theorem 4. $f \succeq^{lsug} g \iff E^{lsug}(f) \geq E^{lsug}(g)$

It should be noticed that $E^{lsug}(1_L A 0_L) = \chi^*(\gamma(A))$ and more generally that:

$$\forall x \succ_P y : E^{lsug}(xAy) \geq E^{lsug}(xBy) \begin{array}{l} \iff xAy \succeq^{lsug} xBy \\ \iff \gamma(A) \geq \quad \gamma(B) \end{array} \tag{10}$$

i.e. when utility degrees are Booleans, the comparison of events in the sense of E^{lsug} is perfectly equivalent to the one in terms of γ — that is why we say that this refinement preserves the capacity. More generally, the procedure is perfectly unbiased in the sense that the original information, i.e. the ordinal evaluation of the likelihood of the events on L and the one of the utility degrees of the consequence on the same scale is preserved.

As expected, $\succeq^{lsug}$ is ordinally equivalent to a Choquet integral, namely the one based on the utility $u' = \chi^* \circ u$ and the capacity $\nu = \chi^* \circ \gamma$.

Theorem 5. $f \succeq^{lsug} g \iff Ch_{\chi^* \circ \gamma, \chi^* \circ \mu}(f) \geq Ch_{\chi^* \circ \gamma, \chi^* \circ \mu}(g)$ *where*

$$Ch_{\chi^* \circ \gamma, \chi^* \circ \mu}(f) = \sum_{\alpha_i \in L} \chi^*(\alpha_i) \cdot (\chi^*(\gamma(F_{\alpha_i})) - \chi^*(\gamma(F_{\alpha_{i-1}})))$$

The intuition behind this result is that the ranking of acts is not modified when replacing $\chi^*(\gamma(\alpha_i))$ by $\chi^*(\gamma(\alpha_i)) - \chi^*(\gamma(\alpha_{i+1}))$ in the definition of E^{lsug} (equation 9) since when $\gamma(\alpha_{i+1})$ is negligible with respect to $\gamma(\alpha_i)$. We thus get the Choquet integral.

Corollary 1. *$\succeq^{lsug}$ satisfies the comonotonic Sure Thing Principle.*

It can be noticed that when the capacity is a possibility measure Π (resp. a necessity measure N), $\succeq^{lsug}$ does not recover the ranking of acts provided by expected utility EU^+ (resp. EU^-). The procedures share the same principles, but they are different. The ordering $\succeq^{lsug}$ is $\succeq_{lmax(\succeq_{lmin})}$ applied to a $(l+1)\times 2$ matrix, whose lines are pairs $(\alpha,\ \Pi(F_\alpha))$. On the other hand the definition of EU^+ is based on to an expression tailored to possibility measure ($S_{\Pi,u}(f) = \max_s \min(u(f(s)), \pi(s))$). EU^+ thus applies $\succeq_{lmax(\succeq_{lmin})}$ to an $|S|\times 2$ matrix, whose lines are pairs $(\pi(s), \mu(f(s)))$.

The contrast between the two approaches appears clearly when comparing binary acts $\top A\bot$. $\succeq^{lsug}$ considers that the information about the likelihood of events contained capacity should be respected, and indeed $\top A\bot \succeq^{lsug} \top B\bot$ iff $\Pi(A) \geq \Pi(B)$. On the contrary, $\succeq^{EU^+}$ does not preserve this order over events but refines it: the possibility ordering of event becomes a big-stepped probability ordering.

3.3 A State-Based Refinement

As suggested in the previous section, different formulations of the Sugeno integral may lead to different refinements. For instance, the idempotence of min and max enable the following expression:

$$S_{\gamma,\mu}(f) = \max_{s\in S} \min(\mu(f(s)), \gamma(F_{\mu(f(s))}))$$

We can then use the χ^* transformation (again, with $N \geq |S|$).

$$E^{lstates}(f) = \sum_s \chi^*(\mu(f(s))) \cdot \chi^*(\gamma(F_{\mu(f(s))})) \tag{11}$$

The following equivalent formulation is more practical.

$$E^{lstates}(f) = \sum_i |\{s, \mu(f(s)) = \alpha_i\}| \cdot \chi^*(\alpha_i) \cdot \chi^*(\gamma(F_i)) \tag{12}$$

Let $\succeq^{lstates}$ be the preference ordering induced by $E^{lstates}$. It is a refinement of $\succeq^{sug}$ and does generally not satisfy the STP. But it satisfies the comonotonic STP. Interestingly, it holds that if $\succeq^{sug}$ satisfies WSTP, then $\succeq^{lstates}$ satisfies STP.

$E^{lstates}$ induces an order on binary acts that can be different from the one encoded by γ. Indeed, $E^{lstates}(\top A\bot) = |A|.\chi(\gamma(A))$. So, $A \sim^{lstates} B \iff (\gamma(A) = \gamma(B)$ and $|A| \sim |B|)$, and $A \succ^{lstates} B \iff (\gamma(A) > \gamma(B)$ or $(\gamma(A) = \gamma(B)$ and $|A| > |B|)$. We get the refinement of the original ranking of events according to γ by the ranking in terms of cardinality.

Moreover, it turns out that $\succeq^{lsug}$ and $\succeq^{lstates}$ are not comparable: $\succeq^{lstates}$ is not a refinement of $\succeq^{lsug}$, nor is $\succeq^{lsug}$ a refinement of $\succeq^{lstates}$, as shown by the following counter example.

Example 1

Let

	f	g	π
s_1	0.8	0.8	0.8
s_2	0.6	0.8	0
s_3	0.7	0.6	1

Then $S_{\gamma,\mu}(f) = S_{\gamma,\mu}(g) = 0.8$. *The vectors of pairs* $(\alpha_i, \gamma(F_i))$ *are:* $((0.8, 0.8), (0.7, 1), (0.6, 1))$ *for* f *and* $((0.8, 0.8), (0.7, 0.8), (0.6, 1))$ *for* g. *Hence* $E^{lsug}(f) > E^{lsug}(g)$. *Now use pairs* $(\mu(f(s)), \gamma(A_{\mu(f(s))})$. *We get:* $((0.8, 0.8), (0.7, 1), (0.6, 1))$ *for* f *and* $((0.8, 0.8), (0.8, 0.8)(0.6, 1))$ *for* g. *Hence* $E^{lstates}(g) > E^{states}(f)$. E^{lsug} *and* $E^{lstates}$ *make opposite rankings, hence one cannot refine the other.*

On this example, the choice of E^{lsug} is closer to the intuition than the one of E^{states}, because g is better than f only on an impossible state while f is as least as good as g on each non-impossible state. Clearly, the problem with E^{states} is that impossible states may influence the decision.

3.4 Refinement Using Moebius Transforms

Another approach to the same problem may start from the expression of Sugeno integral involving all subsets of S:

$$S_{\gamma,\mu}(f) = \max_{A \subseteq S} \min(\gamma^{\#}(A), u_A(f)) \tag{13}$$

where $u_A(f) = \min_{s \in A} u(f(s))$ and $\gamma^{\#}(A) = \gamma(A)$ if $\gamma(A) > \max_{B \subsetneq A} \gamma(B)$, and 0_L otherwise. $\gamma^{\#}$ is the qualitative Moebius transform of γ [2]. The above expression of the Sugeno integral has the standard maxmin form w.r.t. a possibility distribution (on the power set of S). Consider the increasing transformation χ^* that changes a max-min aggregation into a sum-of-products encoding of its $Leximax(\succeq_{lmin})$ refinement

$$EU^{lex\#}(f) = \sum_{A \in 2^S} \chi^*(u_A(f)) \cdot \chi^*(\gamma^{\#}(A)) \tag{14}$$

Notice that here the referential is neither S nor L, but 2^S; so, in the definition of χ^*, we set $N = 2^{Card(S)}$. We normalize the transformation in such a way that $\sum_{A \in 2^S} \chi^*(\gamma^{\#}(A)) = 1$. So, the function $m_* : 2^S \mapsto [0, 1]$:

$$m_*(A) = \chi^*(\gamma^{\#}(A))$$

is a positive mass assignment. Note that m_* is a big-stepped mass function in the sense that:

$$m_*(A) > 0 \implies m_*(A) > \sum_{B \subseteq S, \text{ s. t.} m_*(B) < m_*(A)} m_*(B). \tag{15}$$

[2] It is a qualitative counterpart of the following expression of the Choquet integral: $Ch_\nu(f) = \sum_{A \subseteq S} m_\nu(A) \times u_A(f)$
where m_ν is the Moebius transform of a numerical capacity ν.

In particular, if $\gamma^{\#}(A) > 0_L$ then $m_*(A) > \max_{B \subsetneq A} m_*(B)$. Now, it is easy to show that $\chi^*(u_A(f)) = \chi^*(\min_{s\in A} u(f(s))) = \min_{s\in A} \chi^*(u(f(s)))$. Then:

$$EU^{lex\#}(f) = \sum_{A\subseteq S} m_*(A) \ . \ \min_{s\in A} \chi^*(u(f(s))) \tag{16}$$

is a Choquet integral w.r.t. a belief function. Letting $Bel_*(A) = \sum_{B\subseteq A} m_*(B)$ be the induced belief function, the obtained Choquet integral reads:

$$EU^{lex\#}(f) = Ch_{Bel_*}(f) = \sum_{i=0,\ldots,m-1} \chi(\alpha_i).(Bel_*(F^{\alpha_{i+1}}) - Bel_*(F^{\alpha_i})) \tag{17}$$

This shows that any Sugeno integral can be refined by a Choquet integral *w.r.t a belief function*. In summary:

Theorem 6. *For any Sugeno integral $S_{\gamma,\mu}$, there exist a Choquet integral Ch_{Bel_*,u^*} with respect to a belief function Bel_* and a utility function u^* such that:*

$$S_{\gamma,\mu}(f) > S_{\gamma,\mu}(g) \implies Ch_{Bel_*,u^*}(f) > Ch_{Bel_*,u^*}(g)$$

Contrary to the Choquet integral presented in the Section 3.2, the capacity γ is generally not preserved under the present transformation. The resulting Choquet integral is always pessimistic, and sometimes much more, sometimes not more refined than the original criterion. Two extreme particular cases are interesting to consider:

- If γ is a possibility measure Π, then $\gamma^{\#}(A)$ is positive on singletons of positive possibility only. In other words, $\gamma^{\#}$ coincides with the possibility distribution of Π and the Moebius expression of the Sugeno integral coincides with the expression of the optimistic possibilistic criterion. So m_* is a regular big-stepped probability and the Choquet integral collapses to the regular expected utility EU^+.
- On the contrary if γ is a necessity measure N, Ch_{Bel_*,u^*} does not collapse at all with the pessimistic expected utility EU^-. Indeed, if γ is a necessity measure N, $\gamma^{\#}(A)$ is positive on alpha-cuts of the possibility distribution only. So the mass assignment m_* is positive on the nested family of sets A_i, and the belief function Bel_* is a necessity measure ordinally equivalent to the original one. In this case, the resulting Choquet integral preserves the necessity measure. Only the "max-min" framing of the Sugeno integral has been turned into a "sum-product" framing: the transformation has preserved the nature of the original capacity and the refinement identified in Section 3.2 is retrieved.

4 Conclusion

This paper tries to bridge the gap between qualitative and quantitative criteria for decision-making under uncertainty with a view to increase their

discrimination power. It is shown that qualitative criteria can be refined by symmetric lexicographic procedures. These procedures can be simulated by quantitative criteria using big-stepped scale transformations that preserve the order-of -magnitude and negligibility phenomena pervading the qualitative criteria. When uncertainty is encoded by means of qualitative possibility and necessity measures, a big-stepped expected utility criterion is known to provide a maximal refinement. Here we study the case when uncertainty is encoded by a general qualitative capacity and the criterion is a Sugeno integral. Our results indicate that a Choquet integral is the natural choice for defining refined rankings.

Numerous questions remain open and are not considered here by lack of space, for instance the detailed study of the refinements of a capacity and the question of the (non) unicity of the maximal refinement. Can the capacity-preserving refinement be improved by refining the capacity in a second step? How to relate the various refinements obtained by the various expressions of Sugeno integral? Lastly, finding complete act-driven axiomatics of the new decision rules proposed here is also in order, by putting together Savage axioms and Sugeno integral axioms in some way.

References

1. Doyle, J., Thomason, R.H.: Background to qualitative decision theory. AI Magazine 20(2), 55–68 (1999)
2. Dubois, D., Prade, H.: Possibility theory as a basis for qualitative decision theory. In: Proceedings of IJCAI'95, pp. 1925–1930 (1995)
3. Dubois, D., Prade, H., Sabbadin, R.: Qualitative decision theory with Sugeno integrals. In: Proceedings of UAI'98, pp. 121–128 (1998)
4. Fargier, H., Sabbadin, R.: Qualitative decision under uncertainty: Back to expected utility. Artificial Intelligence 164, 245–280 (2005)
5. Grabisch, M.: Some lexicographic approaches to the Sugeno integral. In: Proceedings International Conference on Information Processing and Management of Uncertainty (IPMU) (2006)
6. Marichal, J.-L.: An axiomatic approach of the discrete Sugeno integral as a tool to aggregate interacting criteria in a qualitative framework. IEEE transactions on fuzzy systems 9(1), 164–172 (2001)
7. Murofushi, T.: Lexicographic use of Sugeno integrals and monotonicity conditions. IEEE Transactions on Fuzzy Systems 9(6), 783–794 (2001)
8. Sarin, R., Wakker, P.P.: A simple axiomatization of nonadditive expected utility. Econometrica 60(6), 1255–1272 (1992)
9. Savage, L.J.: The Foundations of Statistics. Wiley, New York (1954)
10. Schmeidler, D.: Integral representation without additivity. Prooceedings of the American Mathematical Society 97(2), 255–261 (1986)
11. Sugeno, M.: Theory of fuzzy integral and its applications. PhD thesis, Tokyo Institute of Technology, Tokyo (1974)

Algebraic Structures for Bipolar Constraint-Based Reasoning

Hélène Fargier[1] and Nic Wilson[2]

[1] IRIT, 118 route de Narbonne 31062 Toulouse Cedex, France
[2] Cork Constraint Computation Centre, University College Cork, Ireland
fargier@irit.fr, n.wilson@4c.ucc.ie

Abstract. The representation of both scales of cost and scales of benefit is very natural in a decision-making problem: scales of evaluation of decisions are often bipolar. The aim of this paper is to provide algebraic structures for the representation of bipolar rules, in the spirit of the algebraic approaches of constraint satisfaction. The structures presented here are general enough to encompass a large variety of rules from the bipolar literature, as well as having appropriate algebraic properties to allow the use of CSP algorithms such as forward-checking and algorithms based on variable elimination.

1 The Introduction

Soft constraints frameworks usually consider that preferences are expressed in a negative way. For instance, Valued Constraint Satisfaction Problems (VCSPs, [16]) aim at minimising an increasing combination of the violation costs provided by the constraints. This also is the case for all instances of semiring-based CSPs [3], where the combination of the successive valuations provided by the constraints decreases (worsens) the global evaluation. But problems often also contain positive preference constraints which increase the global satisfaction degree, so it is desirable to extend constraints approaches to such situations. For example, if one is choosing a holiday apartment, one has to balance the (positive) benefits of a decision, such as having a sea view, against the (negative) monetary cost. This bipolar characteristic of the preferences in CSPs has recently been advocated by Bistarelli, Pini, Rossi and Venable [4,14,15]. Bipolarity is also an important focus of research in several domains, e.g. psychology [17,18], multicriteria decision making [8,9], and more recently in AI: argumentation [1] and qualitative reasoning [10,2,7].

There are basically two ways of representing a bipolar notion on a scale. The first one is the so called *univariate model* proposed by Osgood et al [13]. It consists of a scale with a central neutral element ranging from negative values (below the neutral element) to positive values (higher than the neutral element). This kind of model has recently been introduced into constraint programming in [4,14,15]. Unlike this first model the *bivariate model* introduced by Cacioppo and al. [5] (see for instance [9,8,7]) does not use one but two scales; this can

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 623–634, 2007.

be pictured with a horizontal axis encoding the intensity of positive values, and the vertical axis the intensity of the negative ones. Thus the evaluation is not necessarily totally positive nor totally negative, but can have both positive and negative components. The original motivation for such a model comes from the fact that a subject may feel at the same time a positive response and a negative one for the same characteristic of an object. For a house, being close to a bus station is both good (time is saved) and bad (it is noisy).

The aim of the present paper is to provide algebraic structures for the representation of bivariate bipolar rules, in the spirit of the algebraic approaches of constraint satisfaction. This is to enable combinatorial optimisation over expressive languages of constraints, where both costs and benefits can be expressed. The structure should be rich enough to encompass a large variety of rules from the bipolar literature; but it should have appropriate algebraic properties to allow the use of soft CSP algorithms. The next section discusses classes of bipolar decision rules. Section 3 describes our basic algebraic structure and shows how to represent some decision rules from the literature using this; special subclasses are also examined. Section 4 describes bipolar systems of constraints and a forward checking algorithm for optimisation. In Section 5 we define a richer algebraic structure, bipolar semirings, which allows more complex propagation algorithms.

2 Bipolar Decision Rules

The purpose of a bipolar decision making procedure is to provide a comparison relation $\succeq$ between alternatives, given, for each alternative d, a multiset $P(d)$ of positive evaluations and a multiset set $N(d)$ of negative ones. In the context of preference-based CSPs, $N(d)$ corresponds to preference valuations provided by some negative constraints, as in fuzzy CSPs and more generally, semiring-based CSPs, and $P(d)$ corresponds to the positive valuations provided by reward-based constraints. The basic property of bipolar decision processes is that the bigger $P(d)$ (respectively, $N(d)$) is, the better (resp., worse) d is:

$$P(d') \subseteq P(d) \text{ and } N(d') \supseteq N(d) \implies d \succeq d'$$

Cumulative prospect theory [18] adds to this "bimonotonicity" axiom a second principle: $P(d)$ and $N(d)$ must be separately evaluated by means of two functions that provide an overall positive degree $p(d)$ and an overall negative degree $n(d)$. According to bimonotonicity, p should be maximised and n minimised.

2.1 Univariate Models

Such models represent a situation where p and n are on the same scale and the decision strategy can be modelled by computing a *net predisposition*: $\text{NP}(d) = f(p(d), n(d))$, where f is increasing in its first argument and decreasing in its second. Alternatives d are then ranked increasingly with respect to NP. The most famous example is based on an aggregation by a sum:

$$\mathrm{NP}^{+}(d) = p(d) - n(d) = \textstyle\sum_{v \in P(d)} v - \sum_{v \in N(d)} v.$$

Another example is provided by qualitative reasoning [10]:

$$\mathrm{NP}^{qual}(d) = min(p(d), 1 - n(d))$$

where $p(d) = \max_{v \in P(d)} v$ and $n(d) = \max_{v \in N(d)} v$. More generally, we can consider that $p(d)$ and $n(d)$ are obtained by monotonic and associative combinations of the valuations they contain, namely by a pair of t-conorms[1] $(\otimes^{+}, \otimes^{-})$: $p(d) = \bigotimes^{+}_{v \in P(d)} v$ and $n(d) = \bigotimes^{-}_{v \in N(d)} v$. It should be noticed that $\otimes^{+}$ and $\otimes^{-}$ can be different from each other—for some subjects, their combination of positive effects is more or less isomorphic to a sum, while for the negative scale, the worst value is taken, i.e. $\otimes^{-} = max$. The NP model thus encompasses more than just the simple additive rule. In [4], net predisposition is generalised to semiring-valued constraints through use of (i) two semirings, one, L^{+}, for representing positive degrees of preference, and the other, L^{-}, for representing negative degrees of preference, equipped with their respective multiplications $\otimes^{+}$ and $\otimes^{-}$ and (ii) an operator $\otimes$ defined within $L^{+} \cup L^{-}$ for combining positive and negative elements. The framework then aims at maximising $(\bigotimes^{+}_{v \in P(d)} v) \otimes (\bigotimes^{-}_{v \in N(d)} v)$.

2.2 Bivariate Models

Since they are fundamentally single-scaled, univariate models are not well suited to the representation of all decision making situations. For instance, a conflicting set whose strongest positive argument is equally strong as its strongest negative argument is often difficult to rank (see e.g. [17]). Since a univariate model aggregates a positive and a negative value into either a positive or a negative value, and since such scales are totally ordered, it cannot account for situations of incomparability. Hence the necessity of bivariate models as first proposed by [5] (see also [8]). As discussed in the introduction, a second reason is the necessity of taking into account arguments that have both a positive and a negative aspect. Classical examples of such rules are provided by Pareto rules. In contrast to net predisposition, these do not make any aggregation of p and n, but rather consider that each of the two dimensions is a criterion and that the scales of the criteria are not commensurate. Decision is then made on the basis of a Pareto comparison:

$$\text{Pareto: } d \succeq d' \iff p(d) \geq p(d') \text{ and } n(d) \leq n(d')$$

Letting $\otimes^{+} = \otimes^{-} = max$, one recovers the qualitative rule proposed in [7], but once again, $\otimes^{+}$ and $\otimes^{-}$ can be two different t-conorms.

The Pareto ordering is obviously rather weak, and it is natural to strengthen it by adding extra orderings to represent tradeoffs. For example, in a Pareto

[1] A t-conorm is an increasing associative and commutative operation on some ordered scale $L = [0_L, 1_L]$ with 0_L as unit element and 1_L as absorbing element. We formulate here the rules in the way they apply in a constraint-based setting. Some of them admit a more general definition accounting for non-independent arguments.

system with both scales being $\{0, 1, 2, \dots\} \cup \{\infty\}$, and both combinations being addition, we might add extra orderings such as $(1, 3) \succeq (0, 0)$. The new ordering $\succeq$ is then defined to be the smallest transitive relation which (i) extends both the Pareto ordering and the extra orderings and (ii) satisfies the property that $\otimes$ is monotone over $\succeq$ (see Definition 1 below). In this example we could deduce using the monotonicity property also that $(2, 4) \succeq (1, 1)$. Other instances of the bivariate model in [5] are provided by qualitative reasoning, namely the order of magnitudes formalism in [19] and the "bilexi" qualitative rule in [7].

3 Bipolar Valuation Structures

The constituent elements of a bipolar framework should be a set of valuations A containing a subset of positive valuations (say, A^+), and a set of negative valuations (say, A^-), a combination operator $\otimes$ and a comparison relation[2] $\succeq$ on A. $\succeq$ is a partial order (i.e., $\succeq$ is reflexive, antisymmetric and transitive, but need not be complete).

A^- contains a worst element, say $\bot$ (which could be received upon the violation of some hard constraint), and A^+ contains a best element $\top$ (which could be received upon the ideal satisfaction of the goal(s)). Both share the *neutral* or "indifferent" valuation, that should not modify the evaluation of a decision.

We also will need algorithms for optimisation in the combinatorial case, e.g. branch and bound algorithms. This further restricts the algebraic framework we are looking for: $\otimes$ should not be sensitive to the order in which the constraints are considered, so is assumed to be commutative and associative; it also should be monotonic w.r.t. $\succeq$.

3.1 Definition and Basic Properties

Definition 1. *A bipolar valuation structure is a tuple $\mathcal{A} = \langle A, \otimes, \succeq \rangle$ where:*
- *$\succeq$ is a (possibly partial) order on A with a unique maximum element $\top$ and a unique minimum element $\bot$ (so for all $a \in A$, $\top \succeq a \succeq \bot$);*
- *$\otimes$ is a commutative and associative binary operation on A with neutral element $\mathbf{1}$ (for all $a \in A$, $a \otimes \mathbf{1} = a$); furthermore $\otimes$ is monotone over $\succeq$: if $a \succeq b$ then for all $c \in A$, $a \otimes c \succeq b \otimes c$.*

Notice that the assumption of the existence of elements $\top$ and $\bot$ is not restrictive. If A does not contain them then we can add them whilst maintaining the properties of $\otimes$, $\succeq$, $\mathbf{1}$.

An element a is said to be *positive* if $a \succeq \mathbf{1}$, and it is said to be *negative* if $a \preceq \mathbf{1}$. We write the set of positive elements of A as A^+, and the set of negative elements as A^-. The following proposition gives some basic properties.

Proposition 1. *Let $\mathcal{A} = \langle A, \otimes, \succeq \rangle$ be a bipolar valuation structure. Then*
(i) $\otimes$ is increasing (resp. decreasing) with respect to positive (resp. negative) elements: if $a \in A$ and $p \succeq \mathbf{1} \succeq n$ then $a \otimes p \succeq a \succeq a \otimes n$;

[2] Given a relation $\succeq$, we use $\succ$ to mean the strict part of $\succeq$, so that $a \succ b$ if and only if $a \succeq b$ and $b \not\succeq a$ (i.e., $b \succeq a$ does not hold).

(ii) $\bot$ *(resp.* $\top$*) is an absorbing element in* A^- *(resp.* A^+*);*
(iii) for all $p \in A^+$ *and* $n \in A^-$, $p \otimes n \succeq n$ *and* $p \succeq p \otimes n$, *so that* $p \otimes n$ *is between* n *and* p.

(i) is a key property for bipolar systems, related to bimonotonicity mentioned above. The third property follows from (i) and means that $p \otimes n$ is somewhere between p and n—but it does not imply that $p \otimes n$ is either positive or negative. It may happen that neither $p \otimes n \succeq \mathbf{1}$ nor $\mathbf{1} \succeq p \otimes n$: the set A can contain more elements than purely positive and purely negative ones, which gives it the ability to represent conflicting values that have both a positive and a negative component.

Define A^{PN} to be the set of all those elements of A which can be written as a combination of a positive and a negative element, i.e.,

$$A^{PN} = \{a \in A : a = p \otimes n, \quad n \preceq \mathbf{1} \preceq p\}.$$

A^{PN} contains A^+, A^- and all the valuations that are obtained by combining positive and negative values: it is the core of the bipolar representation.

Proposition 2. *Let* $\mathcal{A} = \langle A, \otimes, \succeq \rangle$ *be a bipolar valuation structure, then* A^-, A^+ *and* A^{PN} *are each closed under* $\otimes$. *Moreover,* A^{PN} *contains* $A^+ \cup A^-$, *including* $\mathbf{1}, \bot$ *and* $\top$. *Hence* $\langle A^{PN}, \otimes, \succeq \rangle$ *is also a bipolar valuation structure.*

Definition 2
A bipolar structure is bivariate *iff* $A = A^{PN}$. *It is* univariate *iff* $A = A^+ \cup A^-$.

In particular, in a univariate system, the combination of a positive element and a negative element is always comparable to the neutral element.

The framework of bipolar valuation structures is general enough to allow valuations outside A^{PN}, but they do not have such a simple interpretation in terms of positive and negative values. Since we are interested in the representation of bipolarity, we focus the paper on bivariate systems (which includes univariate systems).

3.2 Examples

Additive net predisposition For representing NP^+ we will use $\mathcal{A} = \mathbb{R} \cup \{-\infty, +\infty\}$ with $\otimes = +$ and $\succeq = \geq$. So, the neutral element $\mathbf{1}$ equals 0, $A^+ = \mathbb{R}^+ \cup \{+\infty\}$, $A^- = \mathbb{R}^- \cup \{-\infty\}$. We define $-\infty \otimes +\infty = -\infty$, since getting a conflict is very uncomfortable and should be avoided. However, in practice, $+\infty$ is never allocated by any constraint.

Pareto: $Pareto^{\otimes^-, \otimes^+}$ denotes any Pareto rule built from two t-conorms $\otimes^-$ and $\otimes^+$, respectively in $L^- = [0^-, 1^-]$ and $L^+ = [0^+, 1^+]$. The combination is performed pointwise (using the two conorms) and pair (n, p) is preferred to (n', p') if and only if it is better on each co-ordinate. The encoding of such a rule is done using the a product structure: $\mathcal{A} = \langle L^- \times L^+, (\otimes^-, \otimes^+), \succeq^{par} \rangle$ with $\mathbf{1} = (0^-, 0^+), \bot = (1^-, 0^+), \top = (0^-, 1^+)$ where $\succeq^{par}$ is simply defined

by the Pareto principle: $(n,p) \succeq^{par} (n',p') \iff n \leq n'$ and $p \geq p'$. As a particular case, the qualitative $Pareto^{max}$ rule corresponds to the structure $\mathcal{A} = \langle [0,1] \times [0,1], (\max, \max), \succeq^{par} \rangle$ with $\mathbf{1} = (0,0)$, $\bot = (1,0)$, $\top = (0,1)$.

Additive net prediposition is obviously univariate. The rules of the form $Pareto^{\otimes^+,\otimes^-}$ are not univariate but bivariate. So also is the following rule.

Order of magnitude calculus (OOM): In the system of order of magnitude reasoning described in [19], the elements are pairs $\langle \sigma, r \rangle$ where $\sigma \in \{+,-,\pm\}$, and $r \in \mathbb{Z} \cup \{\infty\}$. The system is interpreted in terms of "order of magnitude" values of utility, so, for example, $\langle -, r \rangle$ represents something which is negative and has order of magnitude K^r (for a large number K). Element $\langle \pm, r \rangle$ arises from the sum of $\langle +, r \rangle$ and $\langle -, r \rangle$. $\langle \pm, r \rangle$ can be thought of as the interval between $\langle -, r \rangle$ and $\langle +, r \rangle$, since the sum of a positive quantity of order K^r and a negative quantity of order K^r can be either positive or negative and of any order less than or equal to r. Let $A_{oom} = \{\langle \pm, -\infty \rangle\} \cup \{\langle \sigma, r \rangle : \sigma \in \{+,-,\pm\}, r \in \mathbb{Z} \cup \{+\infty\}\}$. We write also $\langle -, +\infty \rangle$ as $\bot$, and $\langle +, +\infty \rangle$ as $\top$.

The interpretation leads to defining $\otimes$ by: $\langle \sigma, r \rangle \otimes \langle \sigma', r' \rangle = \langle \sigma, r \rangle$ if $r > r'$; it's equal to $\langle \sigma', r' \rangle$ if $r < r'$; and is equal to $\langle \sigma \oplus \sigma', r \rangle$ if $r = r'$, where $\oplus$ is given by: $+ \oplus + = +$ and $- \oplus - = -$, and otherwise, $\sigma \oplus \sigma' = \pm$. Operation $\otimes$ is commutative and associative with neutral element $\langle \pm, -\infty \rangle$. $\succeq$ is defined by the following instances:[3] (i) for all r and s, $\langle +, r \rangle \succeq \langle -, s \rangle$; (ii) for all $\sigma \in \{+,-,\pm\}$, and all r, r' with $r \geq r'$: $\langle +, r \rangle \succeq \langle \sigma, r' \rangle \succeq \langle -, r \rangle$. The relation $\succeq$ is a partial order with unique minimum element $\bot$ and unique maximum element $\top$. The positive elements and the negative elements are both totally ordered, and $A_{oom} = A^{PN}$. However, there are incomparable elements, e.g. $\langle \pm, r \rangle$ and $\langle \pm, s \rangle$ when $r \neq s$.

3.3 Important Subclasses of Bipolar Valuation Structures

Below we discuss some properties and special kinds of bipolar structures.

Unipolar scales: First of all, let us say that $\mathcal{A}$ is purely positive (resp., purely negative) iff $A = A^+$ (resp. $A = A^-$). In such a structure, $\bot = \mathbf{1}$ (resp. $\top = \mathbf{1}$). The most classical example is provided by semiring-based CSPs where $A = A^-$, while purely positive preference structures are considered in [4].

Totally ordered scales: In most of the bipolar rules encountered in the literature, $\succeq$ is complete on $A^+ \cup A^-$, e.g. NP^+, $Pareto^{max,max}$ and A_{oom}. Unless the structure is univariate, this does not imply that $\succeq$ is complete, but that the restriction of $\otimes$ on A^+ (resp. A^-) is a t-conorm (resp. a t-norm).

Strict monotonicity: $\mathcal{A} = \langle A, \otimes, \succeq \rangle$ is said to be *strictly monotonic* if for all $a, b \in A$ and for all $c \neq \top, \bot$, we have $a \succ b \Rightarrow a \otimes c \succ b \otimes c$. Qualitative rules based on max and min operations are not strictly monotonic, while addition-based frameworks often are. Failure of strict monotonicity corresponds to the

[3] This definition is slightly stronger than the one in [19], which doesn't allow $\langle +, r \rangle \succeq \langle \pm, r \rangle \succeq \langle -, r \rangle$; either order can be justified, but our choice has better computational properties.

well known "drowning effect": without strict monotonicity, it may happen that a decision d is not necessarily strictly preferred to d' even though it is strictly preferred to d' by all constraints apart from one that judges them equally.

Idempotent structures: An element $a \in A$ is said to be *idempotent* if $a \otimes a = a$, and $\otimes$ is said to be idempotent if every element of $\mathcal{A}$ is idempotent. The idempotence of $\otimes$ is very useful for having simple and efficient constraint propagation algorithms. Idempotence, which is at work e.g. in $Pareto^{max,\max}$, A_{oom} and in many unipolar structures (e.g. fuzzy CSPs), induces the drowning effect. Naturally, idempotence and strict monotonicity are highly incompatible properties. The range of compatibility of idempotence with a univariate scale is also very narrow—it reduces the structure to a very special form:

Proposition 3. *If bipolar valuation structure $\mathcal{A}$ is idempotent and univariate, then for all $p \in A^+$ and $n \in A^-$, either $p \otimes n = p$ or $p \otimes n = n$.*

Invertibility: The notion of cancellation is captured by the property of invertibility. Element a is said to be *invertible* if there exists element $b \in A$ with $a \otimes b = \mathbf{1}$. A structure is said to be *invertible* if every element in $A - \{\top, \bot\}$ is invertible. $A - \{\top, \bot\}$ then forms a commutative group under $\otimes$. This property is important for the framework in [4,14,15] and fits well with univariate scales. For instance, it is easy to show that when $\succeq$ is complete on A^+, invertibility is a sufficient condition for making a bivariate system univariate.

On the other hand, associativity implies that $\mathbf{1}$ is the only element a which is both idempotent and invertible, since if $a \otimes b = \mathbf{1}$ then $a = a \otimes \mathbf{1} = a \otimes (a \otimes b) = (a \otimes a) \otimes b = a \otimes b = \mathbf{1}$. This means that when $\otimes$ is idempotent a positive argument can never be exactly cancelled by a negative argument: invertibility is strongly related to strict monotonicity.

Proposition 4. *If bipolar valuation structure $\mathcal{A}$ is invertible then it is strictly monotonic.*

This problem is avoided in [4,14,15] by not assuming associativity on their univariate scale. But invertibility should not be considered as a norm, and bivariate systems are generally not invertible.

4 Bipolar Constraints and Optimisation

4.1 Bipolar Systems of Constraints

Let X be a set of variables, where variable $x \in X$ has domain $D(x)$. For $U \subseteq X$, we define $D(U)$ to be the set of all possible assignments to U, i.e., $\prod_{x \in U} D(x)$.

Let $\mathcal{A} = \langle A, \otimes, \succeq \rangle$ be a bipolar valuation structure. An $\mathcal{A}$-*constraint* φ [over X] is defined to be a function from $D(s_\varphi)$ to A, where s_φ, the *scope* of φ, is a subset of variables associated with φ. We shall also refer to φ as a *bipolar constraint*.

Definition 3. *A bipolar system of constraints, over a bipolar valuation structure $\mathcal{A}$, is a triple (X, D, C) where X is a set of variables, D the associated domains and C a multiset of $\mathcal{A}$-constraints over X.*

Bipolar constraint φ allocates a valuation $\varphi(d)$ to any assignment d to its scope. More generally, if d is an assignment to a superset of s_φ, and e is the projection of d to s_φ, then we define $\varphi(d)$ to be $\varphi(e)$. For any assignment d of X, the *bipolar evaluation of* d is $val(d) = \bigotimes_{\varphi \in C} \varphi(d)$.

Many requests can be addressed to a bipolar system of constraints. The most classical one, the optimisation request, searches for one undominated solution: d is undominated if and only if there does not exist any d' such that $val(d') \succ val(d)$. Variants include the search for several (or all the) undominated solutions. The associated decision problem, for a given bipolar valuation structure $\mathcal{A} = \langle A, \otimes, \succeq \rangle$ can be written as:

[BCSP$_\mathcal{A}$]: *Given a bipolar system of constraints over $\mathcal{A}$ and $a \in A$, does there exist an assignment d such that $val(d) \succ a$.*

Proposition 5. *Let $\mathcal{A} = \langle A, \otimes, \succeq \rangle$ be a bipolar valuation structure. Suppose that testing $b \succ a$ is polynomial, and computing the combination of a multiset of elements in A is polynomial. Suppose also that A contains at least two elements. Then BCSP$_\mathcal{A}$ is* NP*-complete.*

Indeed, given these assumptions, for any $\mathcal{A}$, the problem BCSP$_\mathcal{A}$ is in NP, since we can guess assignment d, and test $val(d) \succ a$ in polynomial time. It is NP-hard if A has more than one element since then $\top$ and $\bot$ must be different and so either $\mathbf{1} \neq \bot$ or $\mathbf{1} \neq \top$ (or both); in either case we can use a reduction from 3SAT, by considering bipolar constraints which only take two different values: $\mathbf{1}$ and either $\bot$ or $\top$.

4.2 Forward Checking Algorithm

This section describes a generalization of the Forward Checking algorithm for finding an undominated complete assignment in bipolar systems of constraints. For the sake of brevity, we assume that all the constraints in C are either unary or binary; however, it is not hard to modify the algorithm to be able to deal with constraints of higher arity.

We assume that we have implemented a function $\texttt{UB}(S)$ that, given a finite subset S of A, returns some upper bound of them (with respect to $\succeq$). $\texttt{UB}(S)$ might be implemented in terms of repeated use of a function $\vee(\cdot, \cdot)$ where $\vee(a, b)$ is an upper bound of both a and b (i.e., $\vee(a, b) \succeq a, b$). For example, if least upper bounds exist, we can set $\vee(a, b)$ to be some least upper bound of a and b. In particular, if $\succeq$ is a total order, we can use max. However, very often $\succeq$ is not a total order, e.g., for the Pareto rule. For each constraint φ we write also $\texttt{UB}(\varphi)$ for $\texttt{UB}(\{\varphi(d) : d \in D(s_\varphi)\}$, i.e. an upper bound over the values of φ. $\texttt{UB}(\varphi)$ is an important parameter: in a bipolar system of constraints, future constraints cannot be neglected, since they can increase the current evaluation. In other

terms, setting $\texttt{UB}(\varphi) = \mathbf{1}$ is not sound—while $\texttt{UB}(\varphi) = \top$ is sound but generally inefficient, both because $\top$ is generally not provided by any constraint (nothing is perfect) and because it is not far from being absorbing (and is so on A^+). In practice, for each φ, $\texttt{UB}(\varphi)$ can be pre-computed.

The handling of $\texttt{UB}(\varphi)$ is the main difference between classical Forward Checking and bipolar Forward Checking. The structure of the algorithm is very classical: the top level procedure, `BestSol`, returns global parameter d^*, which will then be an undominated solution, and global parameter b^* which equals $val(d^*)$. The algorithm performs a tree search over assignments, pruning only when there can be no complete assignment below this point with better *val* than the current best valuation b^* (which is initialised as $\bot$).

Without loss of generality, we assume that for each $x \in X$ there exists exactly one unary constraint φ_x on x (if there exists more, we can combine them; if there exists none, we can set $\varphi_x(v) = \mathbf{1}$ for all $v \in D(x)$). The algorithm involves, for each variable x, a unary constraint μ_x, which is initially set to being equal to φ_x. The backtracking is managed with the help of two procedures: $\texttt{StoreDomainsUnary}(i)$ takes a backup copy of the variable domains and the values of the unary constraints μ_x at tree depth i; $\texttt{RestoreDomainsUnary}(i)$ restores them as they were at point $\texttt{StoreDomainsUnary}(i)$.

We write an assignment d to a set of n variables as a set of assignments $x := v$. In particular, $\{\}$ designates the assignment to the empty set of variables.

procedure `BestSol`
$b^* := \bot$
`for all variables` x, `for all` $v \in D(x)$, `set` $\mu_x(v) := \varphi_x(v)$
`FC(0, {}, 1)`
`Return` d^* `and` b^*

procedure $\texttt{FC}(i, d, \mathit{CurrentVal})$
`If` $i = n$ `then`
  `if` $\mathit{CurrentVal} \succ b^*$ `then` $b^* := \mathit{CurrentVal}$; $d^* := d$
`Else`
  `Choose an unassigned variable` x
  $\texttt{StoreDomainsUnary}(i)$
  `For all` v `in` $D(x)$
    `If` $\texttt{PropagateFC}(x, v, \mathit{CurrentVal})$ `then` $\texttt{FC}(i+1, d \cup \{x := v\}, \mathit{CurrentVal} \otimes \mu_x(v))$
    $\texttt{RestoreDomainsUnary}(i)$

boolean function $\texttt{PropagateFC}(x, v, \mathit{CurrentVal})$
$\mathit{PastVal} := \mathit{CurrentVal} \otimes \mu_x(v)$
$\mathit{futureConstr} = \{\varphi$ `linking two unassigned variables`$\}$
$\mathit{FutureVal} := \bigotimes_{\varphi \in \mathit{futureConstr}} \texttt{UB}(\varphi)$
`// Propagate forward on the future variables:`
`For all` φ `linking` x `to an unassigned variable` y
  `For all values` v' `in` $D(y)$
    `set` $\mu_y(v') := \mu_y(v') \otimes \varphi(x = v, y = v')$
  $\mathit{Upper}_y := \texttt{UB}(\{\mu_y(v') : v' \in D(y)\})$

```
// Pruning the domains
For all unassigned variables y and all v' ∈ D(y)
  VarsVal_y := ⊗_{y' unassigned, y'≠y} Upper_{y'}
  UppBd_y(v') := μ_y(v') ⊗ VarsVal_y ⊗ PastVal ⊗ FutureVal
  If not(UppBd_y(v') ≻ b*) remove v' from D(y)
  If D(y) = ∅ then return FALSE (and exit PropagateFC)
Return TRUE
```

The soundness of the pruning condition is ensured by the monotonicity of $\otimes$ and the transitivity of $\succeq$. But it can also be sound in some structures that do not fulfill these conditions. In particular, even if $\otimes$ is not monotone over $\succeq$ then the algorithm will still be correct if operator $\vee$ ensures that $\forall c \in A$, $\vee(a, b) \otimes c$ is an upper bound of $a \otimes c$ and $b \otimes c$.

The family of Forward Checking algorithms includes more complex versions than the one extended here, e.g. Reversible Directional Arc Consistency(RDAC) and other improvements [12]. The algebraic structure presented in Section 3 is rich enough to allow them to work soundly. But more sophisticated algorithms for constraint optimisation, which use more complex constraint propagation (e.g. using variable elimination), require more than a simple upper bound operator. This is the topic of the next section.

5 Bipolar Semirings

An important computational technique for multivariate problems (such as CSPs) is sequential variable elimination (bucket elimination). This calls for the structure to be rich enough to allow the definition of an internal operator $\vee$ that not only provides an upper bound of its operands (and thus admits $\top$ as absorbing element and $\bot$ as a neutral element) but is also assumed to be associative, commutative and idempotent. Unsurprisingly, the kind of structure needed is a semiring, but of a more general form than the semirings usually used in constraint programming. A (commutative) semiring is a set A endowed with two operations $\vee$ and $\otimes$ which are both commutative and associative and such that $\otimes$ distributes over $\vee$.

Definition 4. *A bipolar semiring is a tuple $\langle A, \otimes, \vee, \succeq\rangle$ where: $\langle A, \otimes, \succeq\rangle$ is a bipolar monotonic valuation structure; $\vee$ is an associative, commutative and idempotent operation on A with neutral element $\bot$ and absorbing element $\top$, satisfying:*

Distributivity: for all $a, b, c \in A$, $a \otimes (b \vee c) = (a \otimes b) \vee (a \otimes c)$;
Monotonicity of $\vee$ over $\succeq$: i.e., $a \succeq b \implies a \vee c \succeq b \vee c$.

Notice that, since $a \succeq \bot$, and $\vee$ is monotone over $\succeq$, we have $a \vee b \succeq \bot \vee b = b$. We therefore have the following:

Proposition 6. *Let $\langle A, \otimes, \vee, \succeq\rangle$ be a bipolar semiring. Then for any $a, b \in A$, $a \vee b \succeq a, b$.*

Hence Definition 4 implicitly requires $a \vee b$ to be an $\succeq$-upper bound for a and b, which is an important property for branch-and-bound and variable elimination algorithms. When $\succeq$ is a total order, finding a suitable $\vee$ is immediate: choose $\vee = \max$. When $\succeq$ is an upper semi-lattice, $a \vee b$ will be the least upper bound of a and b. For instance, when A is the product of a totally ordered positive scale and a totally ordered negative scale, as in the Pareto case, we can use pointwise application of maximum. In the OOM framework $\langle \sigma, r \rangle \vee \langle \sigma', r' \rangle$ is the better of the two elements if they are comparable; otherwise, their least upper bound is equal to $\langle +, \max(r, r') \rangle$. It can be shown that $\langle A_{oom}, +, \vee, \succeq, \rangle$ is a bipolar semiring.

Importantly, semiring properties are sufficient for variable elimination to be correct (see e.g., [11]). Hence Definition 4 enables the use of such methods within a branch and bound tree search as a way of generating global upper bounds of a set of bipolar constraints (in particular, as a way to compute a stronger value of *FutureVal* in the above algorithm). However, sequential variable elimination is only practical in certain situations, in particular, if the scopes of the constraints are such that the treewidth is small. Otherwise one can use a mini-buckets [6] approach for generating an upper bound of the least upper bound, since it has been shown that sufficient conditions for this technique to be applicable to general soft constraints, are that $\mathcal{A}$ forms a semiring, the two operators are monotone over the ordering, and $a \vee b \succeq a, b$ for all $a, b \in A$.

Notice that $\vee$ itself defines a comparison relation $\succeq_\vee$ on A, as in semiring-based CSPs: for all $a, b \in A$, $a \succeq_\vee b \iff a \vee b = a$. It follows that for any $a \in A$, we have $\bot \preceq_\vee a \preceq_\vee \top$ and that $\vee$ and $\otimes$ are monotone with respect to $\succeq_\vee$. Hence if $\langle A, \otimes, \vee, \succeq \rangle$ is a bipolar semiring then $\langle A, \otimes, \vee, \succeq_\vee \rangle$ is as well. It is also easy to show that $\succeq_\vee$ is a partial order (it is antisymmetric) but is not necessarily complete. Moreover, by Proposition 6, $a \succeq_\vee b \Rightarrow a \succeq b$. Hence if is optimal (i.e. non dominated) with respect to $\succeq$ then it is optimal w.r.t. $\succeq_\vee$.

6 Conclusion

The representation of both scales of cost and scales of benefit is very natural in a decision-making problem. We have abstracted the kind of properties assumed in such bipolar reasoning to produce algebraic valuation structures which, firstly, allow the representation of many natural forms of bipolar reasoning, and secondly, have sufficient structure to allow optimisation algorithms. As well as bipolar univariate models, our framework can also represent bivariate models for bipolar reasoning, which allow the kind of incomparability found in many natural systems for reasoning with positive and negative degrees of preference.

This paper has proposed a generalization of the forward checking algorithm for handling the optimization in bipolar structures. This algorithm actually applies to rather general algebraic structures, even to structures similar to bipolar valuation structures but which are not fully monotonic.

Acknowledgements. This material is based partly on works supported by the Science Foundation Ireland under Grant No. 00/PI.1/C075 and also Grant No. 05/IN/I886, and benefitted from the Enterprise Ireland Ulysses travel grant No. FR/2006/36.

References

1. Amgoud, L., Bonnefon, J.F., Prade, H.: An argumentation-based approach to multiple criteria decision. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 10–17. Springer, Heidelberg (2005)
2. Benferhat, S., Dubois, D., Kaci, S., Prade, H.: Bipolar representation and fusion of preferences on the possibilistic logic framework. In: Proc. KR2002, pp. 421–448 (2002)
3. Bistarelli, S., Montanari, U., Rossi, F.: Semiring-based Constraint Solving and Optimization. JACM 44(2), 201–236 (1997)
4. Bistarelli, S., Pini, M.S., Rossi, F., Venable, K.B.: Positive and negative preferences. In: Proc. CP 2005 workshop on preferences and soft constraints (Soft 2005) (2005)
5. Cacioppo, J.T., Gardner, W.L., Berntson, G.G.: Beyond bipolar conceptualizations and measures: The case of attitudes and evaluative space. Personality and Social Psychology Review 1, 3–25 (1997)
6. Dechter, R., Rish, I.: Mini-buckets: A general scheme for bounded inference. J. ACM 50(2), 107–153 (2003)
7. Dubois, D., Fargier, H.: Qualitative decision making with bipolar information. In: KR'06, pp. 286–297 (2006)
8. Grabisch, M., Labreuche, Ch.: Bi-capacities — parts I and II. Fuzzy Sets and Systems 151(2), 211–260 (2005)
9. Greco, S., Matarazzo, B., Slowinski, R.: Bipolar Sugeno and Choquet integrals. In: Proc. EUROFUSE 2002 Workshop on Information Systems (2002)
10. Kaci, S., Prade, H.: Bipolar goals. a possibilisitc logic characterization of preferred choices. In: Proc. ECAI'04 Workshop on local computation for logics and uncertainty, pp. 13–18 (2004)
11. Kohlas, J.: Valuation algebras induced by semirings. Technical Report 04-03, Department of Informatics, University of Fribourg (2004)
12. Larrosa, J., Meseguer, P., Schiex, T., Verfaillie, G.: Reversible DAC and other improvements for solving max-CSP. In: Proc. AAAI'96, pp. 347–352 (1998)
13. Osgood, C.E., Suci, G.J., Tannenbaum, P.H.: The Measurement of Meaning. University of Illinois Press, Chicago (1957)
14. Pini, M.S., Rossi, F., Venable, K.B., Bistarelli, S.: Bipolar preference problems. In: Proc. ECAI 2006 (poster paper) (2006)
15. Pini, M.S., Rossi, F., Venable, K.B., Bistarelli, S.: Modelling and solving bipolar preference problems. In: Proc. 2006 ERCIM workshop on constraints (2006)
16. Schiex, T., Fargier, H., Verfaillie, G.: Valued constraint satisfaction problems: Hard and easy problems. In: Proc. IJCAI'95, pp. 631–637 (1995)
17. Slovic, P., Finucane, M., Peters, E., MacGregor, D.G.: Rational actors or rational fools? Implications of the affect heuristic for behavioral economics. The Journal of Socio-Economics 31, 329–342 (2002)
18. Tversky, A., Kahneman, D.: Advances in prospect theory: Cumulative representation of uncertainty. Journal of Risk and Uncertainty 5, 297–323 (1992)
19. Wilson, N.: An order of magnitude calculus. In: Proc. UAI'95, pp. 548–555 (1995)

A Revised Qualitative Choice Logic for Handling Prioritized Preferences

Salem Benferhat and Karima Sedki

CRIL CNRS, Université d'Artois, Rue Jean Souvraz 62307-Lens France
{benferhat,sedki}@cril.univ-artois.fr

Abstract. *Qualitative Choice Logic (QCL)* is a convenient tool for representing and reasoning with "basic" preferences. However, this logic presents some limitations when dealing with complex preferences that, for instance, involve negated preferences. This paper proposes a new logic that correctly addresses QCL's limitations. It is particularly appropriate for handling prioritized preferences, which is very useful for aggregating preferences of users having different priority levels. Moreover, we show that any set of preferences, can equivalently be transformed into a set of normal form preferences from which efficient inferences can be applied.

1 Introduction

Decision analysis and Artificial Intelligence have been developed almost separately. Decision analysis is concerned with aggregation schemes and has relied mostly on numerical approaches, while Artificial Intelligence deals with reasoning and has an important logically oriented tradition [5]. Artificial Intelligence methods can contribute to a more implicit and compact representation of "agent's" preferences. This line of research has been recently illustrated in various ways by AI researchers [9,7,12,2].

Recently, a new logic for representing choices and preferences has been proposed [1]. This logic, called *Qualitative Choice Logic (QCL)*, is an extension of propositional logic. The non-standard part of *QCL* logic is a new logical connective $\times$, called *Ordered disjunction*, which is fully embedded in the logical language. Intuitively, if A and B are propositional formulas then A $\times$ B means: "if possible A, but if A is impossible then at least B". As a consequence, *QCL* logic can be very useful to represent preferences for that framework. However, it presents some limitations. Assume that we want to represent the options concerning a travel from Paris to Vancouver. Assume that a travel agency has the following rules "customers preferring Air France to KLM also buy a hotel package" and "customers preferring KLM to Air France do not buy a hotel package". When a travel agency meets a customer that actually prefers Air France to KLM, the expected behavior of its information system is to propose a hotel package to that customer. Unfortunately, the *QCL* logic does not allow us to infer such a conclusion. It will infer both that a package should and should not be proposed. In fact, the way negation is handled in *QCL* logic is not fully satisfactory. In

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 635–647, 2007.

QCL when a negation is used on a QCL formula with ordered disjunctions, that negated QCL formula is logically equivalent to a propositional formula obtained by replacing the ordered disjunction ($\times$) by the propositional disjunction ($\vee$). This is really a limitation, since for instance QCL does not make a distinction between the three rules: "Air France $\times$ KLM $\Rightarrow$ FirstClass" (people preferring Air France to KLM travel in first class), "KLM $\times$ Air France$\Rightarrow$FirstClass" (people preferring KLM to Air France travel in first class) and "Air France $\vee$ KLM $\Rightarrow$ FirstClass" (people flying on Air France or KLM travel in first class). In their two page short paper [4] these limitations of QCL have been informally advocated, however no rigorous solution is proposed.

This paper proposes a new logic called $PQCL$ (Prioritized Qualitative Choice Logic). It is a new logic in the sense that negation, conjunction and disjunction departs from the ones used in standard QCL. However, it is based on the same QCL language. Our logic is dedicated for handling prioritized preferences and its inference relation correctly deals with negated preferences. In many applications, agent's preferences do not have the same level of importance. For instance, an agent who provides the two preferences : " I prefer AirFrance to KLM", and " I prefer a windows seat to a corridor seat", may consider that the first preference statement is more important that the second preference statement. Our logic can manage such prioritized preferences using prioritized conjunction, and disjunction.

One of the strong point of the $PQCL$ logic proposed in this paper, is that its inference relation can be constructed in two equivalent ways. The first way is based on inference rules that define the satisfaction degree for any formula. The second way is based on a normal form function that equivalently transforms any set of preferences into a set of normal form preferences, from which efficient inferences can be applied. Indeed, having a set of preferences in a normal form allows us to reuse various non-monotonic approaches such as possibilistic logic [10] or compilation of stratified knowledge bases [16].

The rest of this paper is organized as follows. First, we recall QCL language, and we describe QCL logic limitations. Then, we introduce our new logic, called $PQCL$, that deals with prioritized preferences. We present the inference relation for our $PQCL$ logic and show how $PQCL$ theories can be transformed into normal form theories. Last section concludes the paper.

2 The QCL language

This section presents the QCL language, which is in fact composed of three encapsulated sub-languages: Propositional Logic Language, the set of Basic Choice Formulas (BCF) or normal form preferences and the set of General Choice Formulas (GCF). These sub-languages are presented in the following subsections.

2.1 Basic Choice Formulas (BCF)

Let PS denotes a set of propositional symbols and $PROP_{PS}$ denotes the set of propositional formulas that can be built using classical logical connectives ($\Leftrightarrow, \Rightarrow, \wedge, \vee, \neg$) over PS.

Basic choice formulas are ordered disjunctions of propositional formulas. They propose a simple way to order available alternatives. Given a set of propositional formulas $a_1, a_2, ..., a_n$, the formula $a_1 \times a_2 \times \ldots \times a_n$[1] is used to express and ordered list of alternatives: some a_i must be true, preferably a_1, but if this is not possible then a_2, if this is not possible a_3, etc.

The language composed of *basic choice formulas* is denoted by BCF_{PS}, is the smallest set of words defined inductively as follow:

1. If $\phi \in PROP_{PS}$ then $\phi \in BCF_{PS}$
2. If $\phi, \psi \in BCF_{PS}$ then $(\phi \times \psi) \in BCF_{PS}$
3. Every *basic choice formula* is only obtained by applying the two rules above a finite number of times.

BCF formulas represent simples alternatives between propositional formulas. In the rest of this paper BCF formulas are also called normal form formulas. The language of *basic choice formulas* has strong relationships with possibility theory, in particular with guaranteed possibility distributions, (see [1] for more details).

2.2 General Choice Formulas (GCF)

General Choice Formulas represent any formula that can be obtained from PS using connectors $\times, \wedge, \vee, \neg$ on propositional formulas. The language composed of *general choice formulas*, denoted by QCL_{PS}, is defined inductively as follows:

1. If $\phi \in BCF_{PS}$ then $\phi \in QCL_{PS}$
2. If $\phi, \psi \in QCL_{PS}$ then $(\phi \wedge \psi), \neg(\psi), (\phi \vee \psi), (\phi \times \psi) \in QCL_{PS}$.
3. The language of QCL_{PS} is only obtained by applying the two rules above a finite number of times.

GCF formulas represent the whole set of formulas that can be built using four connectives $(\times, \wedge, \vee, \neg)$.

Example 1. The formula "AirFrance × KLM" is a Basic Choice Formula, while the formula "(AirFrance ×¬ KLM)∨ (Class 1 × Class 2)" is a General Choice Formula.

For lake of space, we do not recall the inference process for QCL language. See [1] for a full description of QCL.

3 Limitations of QCL

As advocated in the introduction, the original QCL inference relation has a couple of (intuitively) undesirable properties. In the scope of a negation symbol or when occurring in the antecedent of a (material) implication, ordered

[1] The operator $\times$ is associative. Hence, $a_1 \times a_2 \times \ldots \times a_n$ is used as a shorthand for $(((a_1 \times a_2) \times a_3) \ldots \times a_n \ldots)))$.

disjunctions have not got their intended preferential reading. In fact, negated QCL formula is equivalent to plain propositional formula, obtained by replacing ordered disjunction by a standard disjunction, for instance :

$$\neg(a_1 \times a_2 \times \ldots \times a_n) \equiv \neg a_1 \wedge \neg a_2 \wedge \ldots \wedge \neg a_n.$$

This means that, the double negation of any QCL formula is not equivalent to that formula, namely $\neg\neg(\phi)$ is not equivalent to ϕ. As a negative consequence, QCL does not make distinction between the following three rules :

1. $\neg$(AirFrance $\times$ KLM) $\vee$ FirstClass,
2. $\neg$(KLM $\times$ AirFrance) $\vee$ FirstClass,
3. $\neg$(AirFrance $\vee$ KLM) $\vee$ FirstClass.

They are all equivalent to the propositional formula ($\neg$AirFrance$\wedge\neg$KLM)$\vee$ FirstClass.

4 Prioritized Qualitative Choice Logic *(PQCL)*

This section proposes a new logic called *Prioritized Qualitative Choice Logic (PQCL)*, which is characterized by new definitions of negation, conjunction and disjunction that are useful for aggregating preferences of users having different priority levels and overcome the QCL limitations. As in standard propositional logic, an interpretation I is an assignment of the classical truth values T,F to the atoms in PS. I will be represented by the set of its satisfied literals. The main features of our $PQCL$ logic are :

- The semantics of any formula is based on the degree of satisfaction of a formula in a particular model I. If an interpretation I satisfies a formula ϕ, then its satisfaction degree should be unique. We will use the notation $I|\sim_i^{PQCL}\phi$ to express that I satisfies ϕ to a degree i.
- Negation: Negation should be as a close as possible to the one of propositional logic. In particular, a double negation of a given formula should recover the original formula, namely we want $\neg(\neg\phi)$ to be equivalent to ϕ.
  Note that one cannot simply define $I|\sim_i\neg\phi$ iff "$I|\sim_i\phi$ is not true". Indeed, this implies that the satisfaction degree of a negated formula is not unique (which is not desirable). Namely, if one accepts $I|\sim_i\neg\phi$ iff $I|\sim_i\phi$ is not true, then if a given interpretation I satisfies ϕ to a degree 1 (namely, $I|\sim_1\phi$), then this means that $I|\sim_2\neg\phi$ and $I|\sim_3\neg\phi$ are valid (since $I|\sim_2\neg\phi$ and $I|\sim_3\neg\phi$ are not valid), hence $\neg\phi$ is satisfied to different degrees which is not desirable.
  An additional feature is that the negation should be decomposable with respect to the conjunction and the disjunction, and it should satisfy De Morgan law.
- Prioritized preferences: Our $PQCL$ logic should deal with prioritized preferences, encoded by means of a prioritized conjunction (resp. disjunction). For instance, an agent who provides the two preferences : " I prefer AF to

KLM", and " I prefer a Windows seat to a Corridor seat" which we can formulate by "(AF × KLM) ∧(Windows×Corridor)" may consider that (AF × Windows) is more important than (KLM × Corridor).

4.1 The *PQCL* Inference Relation

Before formally defining $\mid\!\sim^{PQCL}$, we need to introduce the notion of optionality, which is a revised version of the one given in [1].

Definition 1 (Optionality). *Let ϕ_1 and ϕ_2 be two formulas in QCL_{PS}. The optionality of a formulas is a function that assigns to each formula a strictly positive integer.*

- *opt(A) = 1 , A is a atom.*
- *opt($\phi_1 \times \phi_2$)= opt(ϕ_1)+ opt(ϕ_2).*
- *opt($\phi_1 \wedge \phi_2$)= opt(ϕ_1)× opt(ϕ_2).*
- *opt($\phi_1 \vee \phi_2$)= opt(ϕ_1)× opt(ϕ_2).*
- *opt($\neg(\phi_1)$) = opt(ϕ_1).*

The optionality of a formula indicates the number of satisfaction degrees that a formula can have. The main difference with the original definition of optionality given in [1] concerns the three last definitions of optionality. In particular, in [1] opt($\neg\phi$) always equals 1 (since $\neg\phi$ is equivalent to a propositional formula). In our new logic $\neg\phi$ has the same optionality as ϕ. Note that the optionality of a propositional formulas is equal to 1.

The justification of optionality degree is directly related to the definition of of satisfaction degrees associated with interpretations. Let us explain, when dealing with prioritized preferences, why there are opt(ϕ_1)× opt(ϕ_2) options to satisfy ($\phi_1 \wedge \phi_2$). First, opt(ϕ_1) (resp. opt(ϕ_2)) means that ϕ_1 (resp. ϕ_2) can be satisfied to a degree 1(first option of ϕ_1), to a degree 2(second option of ϕ_1),...,to a degree opt(ϕ_1)(the last option for ϕ_1), (resp. 1, 2,...,opt(ϕ_2)). Intuitively, given ($\phi_1 \wedge \phi_2$), the best and preferred solution is the one which satisfies the first option of ϕ_1 and the first option of ϕ_2. Then the second preferred solution is the one that still satisfies the first option of ϕ_1, but only satisfies the second option of ϕ_2. And more generally, an interpretation w is preferred to an interpretation w', if :

1. either w satisfies ϕ_1 to a degree i, and w' satisfies ϕ_1 to a degree j with j > i, or
2. both w and w' satisfy ϕ_1 to a same degree, but the degree on which ϕ_1 is satisfied in w is lower than the degree on which ϕ_2 is satisfied in w'.

Clearly, there are opt(ϕ_1)× opt(ϕ_2) options to satisfy $\phi_1 \wedge \phi_2$. And the worst solution is the one which satisfies ϕ_1 to a degree equal opt(ϕ_1) and ϕ_2 to a degree equal opt(ϕ_2).

Given these optionality degrees associated with formulas, we are now able to define $\mid\!\sim^{PQCL}$. This is given by the following definition.

Definition 2 (The satisfaction relation). *Let ϕ_1, ϕ_2 be two formulas from QCL_{PS}. Let I be an interpretation. The following items give the definition of a satisfaction degree k of a formula ϕ_1 by I, denoted by $I \mid\!\sim_k^{PQCL} \phi_1$.*

1. $I|\sim_k^{PQCL}$ a iff $k = 1$ and $a \in I$ (for propositional atoms a).
2. $I|\sim_k^{WQCL}\neg a$ iff $k = 1$ and $\neg\ a \in I$ (for propositional atoms a).
3. $I|\sim_k^{PQCL}(\phi_1 \times \phi_2)$ iff *($I|\sim_k^{PQCL}\phi_1$)* or *($I|\sim_n^{PQCL}\phi_2$* and there is no m such that $I|\sim_m^{PQCL}\phi_1$, and $k = n$ + *opt(ϕ_1))*.
4. $I|\sim_k^{PQCL}(\phi_1 \vee \phi_2)$ iff one of the following cases is satisfied *:*
   (a) ($I|\sim_1^{PQCL}\phi_1$) or *($I|\sim_1^{PQCL}\phi_2$)* and k = 1.
   (b) (There exists $i > 1$, $I|\sim_i^{PQCL}\phi_1$) and [$\nexists$ m such that $I \mid\sim_m^{PQCL}\phi_2$], and k = (i-1) × opt($\phi_2$) + 1.
   (c) (There exists $i > 1$ such that $I|\sim_i^{PQCL}\phi_1$ or $\nexists$ l, such that $I|\sim_l^{PQCL}\phi_1$) and *(there is $j > 1$, $I|\sim_j^{PQCL}\phi_2$)*, and $k = j$.
5. $I|\sim_k^{PQCL}(\phi_1 \wedge \phi_2)$ iff $I|\sim_i^{PQCL}(\phi_1)$ and $I|\sim_j^{PQCL}(\phi_2)$ and k =*(i-1)*×*opt(ϕ_2)* + j.
6. $I|\sim_k^{PQCL}\neg(\phi_1 \vee \phi_2)$ iff $I|\sim_k^{PQCL}\neg\phi_1 \wedge \neg\phi_2$.
7. $I|\sim_k^{PQCL}\neg(\phi_1 \wedge \phi_2)$ iff $I|\sim_k^{PQCL}\neg\phi_1 \vee \neg\phi_2$.
8. $I|\sim_k^{PQCL}\neg(\phi_1 \times \phi_2)$ iff $I|\sim_k^{PQCL}\neg\phi_1 \times \neg\phi_2$.
9. $I|\sim_k^{PQCL}\neg(\neg\phi_1)$ iff $I|\sim_k^{PQCL}\phi_1$.

Let us explain the definition of our inference relation. Items (2), (6), (7), (8), (9) deals with the satisfaction of negated formulas. They simply say that the negation is decomposable, and satisfies De Morgan rule. Items (5) and (6) deals with the satisfaction of conjunction and the satisfaction of disjunction. Clearly, they deal with prioritized preferences. The way the degree is defined on $\phi_1 \wedge \phi_2$ reflects some lexicographical ordering between individual satisfaction degree of ϕ_1 and ϕ_2. Namely, given two interpretations I, I' then I is strictly preferred to I', if :

1. either $I|\sim_i\phi_1$ and $I'|\sim_j\phi_1$ with $i < j$,
2. or ($I|\sim_i\phi_1$ and $I'|\sim_i\phi_1$) and ($I|\sim_k\phi_2$ and $I'|\sim_l\phi_2$ with $k < l$).

Example 2. Let us illustrate the inference relation of $PQCL$ by the following example. Let $\phi = (AF \times KLM) \wedge (W \times C)$, and I = {KLM,C}. The formula ϕ is of the form $(\phi_1 \wedge \phi_2)$ with $\phi_1 = (AF \times KLM)$ and $\phi_2 = W \times C$.

We have $I|\sim_{i=2}^{PQCL}\phi_1$ and $I|\sim_{j=2}^{PQCL}\phi_2$ (by using item (3) of Definition 2). Thus, applying item (5) of Definition 2, we obtain $I|\sim_k^{PQCL}\phi$ and k = (i-1)×opt(ϕ_2)+j = 4.

Comparing to the original QCL inference relation defined in [1], only the item (1) and the item (3) are the same. All others are different. Namely, our definition of negation, conjunction and disjunction are completely different. However, there is situation where QCL and $PQCL$ collapse. It is when restricting to propositional or basic choice formulas, more precisely :

Proposition 3. *1. Let ϕ be a propositional formula, and I an interpretation, then : $I|\sim_1^{QCL}\phi$ iff $I|\sim_1^{PQCL}\phi$ iff $I \models \phi$.*
2. Let $\phi = a_1 \times a_2 \times \ldots \times a_n$ be a basic choice formula namely, a_i are propositional formulas, then : $I|\sim_k^{QCL}\phi$ iff $I|\sim_k^{PQCL}\phi$ iff $I \models a_1 \vee a_2 \vee \ldots \vee a_n$ and $k = min\{j \mid I \models a_j\}$.

Namely, a basic choice formula $a_1 \times a_2 \times \ldots \times a_n$ is satisfied to a degree k by an interpretation I if I satisfies a_k but fails to satisfy a_i for all $1 \leq i < k$.

The above proposition shows that our definitions of disjunction and conjunction extend the ones of classical logic when they are applied to propositional formulas. But of course they are non-classical since they can be used on non-propositional (general QCL) formulas.

Now, we define an inference relation between a $PQCL$ theory and a propositional formula. It follows the same steps defined in [1]. Let K be a set of propositional formulas which represents knowledge or integrity constraints, and let T be a set of preferences. We need to define the notion of preferred models.

Definition 4. *Let $M^k(T)$ denote the subset of formulas of T satisfied by a model M to a degree k. A model M_1 is $K \cup T$-preferred over a model M_2 if there is a k such that $\mid M_1^k(T) \mid > \mid M_2^k(T) \mid$ and for all $j < k$: $\mid M_1^j(T) \mid = \mid M_2^j(T) \mid$. M is a preferred model of $K \cup T$ iff :*

1. *M is model of K, and*
2. *M is maximally $(K \cup T)$-preferred.*

The following definition gives the inference relation between $(K \cup T)$ and a propositional formula ϕ.

Definition 5. *Let K be a set of formulas in $PROP_{PS}$ and T be a set of formulas in GCF_{PS}, and ϕ be a formula in $PROP_{PS}$.*
$K \cup T \mid\sim_k^{PQCL} \phi$ iff ϕ is satisfied in all preferred models of $K \cup T$.

The following contains an example which shows that $PQCL$ overcomes one the limitations of QCL advocated in the introduction. Note that if one wants to express that ϕ_1, ϕ_2 are equally really, then it is en-ought to separately put them in T.

Example 3. Let us consider the example given in the introduction where our knowledge base K contains $\neg$KLM $\vee \neg$AirFrance **(1)** and T contains the following preferences:

$$\begin{cases} \neg(AirFrance \times KLM) \vee HotelPackage \quad (\mathbf{2}) \\ \neg(KLM \times AirFrance) \vee \neg HotelPackage \quad (\mathbf{3}) \\ AirFrance \times KLM \quad (\mathbf{4}) \end{cases}$$

To give the preferred model of $K \cup T$, we should firstly give the satisfaction degree of the formulas **(1)**, **(2)**, **(3)** and **(4)** for each interpretation, so for instance, by using Definition 2. Let $I = \{$AirFrance, hotelpackage$\}$, We have $I \models \neg$KLM $\vee \neg$AirFrance, $I \mid\sim_1^{PQCL} AirFrance \times KLM$ (using Proposition 1).

Consider now the preference **(2)**, namely $\neg(AirFrance \times KLM) \vee Hotel\ Package$. Using item (8) of Definition 2, we get $I \mid\sim_2^{PQCL} \neg(AirFrance \times KLM)$, and $I \mid\sim_1^{PQCL} HotelPackage$. Using item (4)-a of Definition 2, we get $I \mid\sim_1^{PQCL} (\neg AirFrance \times KLM) \vee HotelPackage$. Similarly, we also have $I \mid\sim_1^{PQCL} \neg(KLM \times AirFrance) \vee \neg HotelPackage$, since $I \mid\sim_1^{PQCL} \neg(KLM \times AirFrance)$.

Table 1. The models of $K \cup T$ by using $PQCL$

AirFrance	KLM	Hotelpackage	**(1)**	**(2)**	**(3)**	**(4)**
F	F	F	1	1	1	-
F	F	T	1	1	1	-
F	T	F	1	1	1	2
F	T	T	1	1	2	2
T	F	F	1	2	1	1
T	**F**	**T**	**1**	**1**	**1**	**1**
T	T	F	-	-	1	1
T	T	T	-	1	-	1

The following truth table summarizes for each formula **(1)**, **(2)**, **(3)**, **(4)** from K and T we are interested in, whether it is satisfied (T) (to some degree) or not (-) by a given interpretation.

In original QCL logic, $K \cup T$ is declared to be inconsistent. With our $PQCL$ logic, $K \cup T$ has one preferred model (bold line), $I = \{AirFrance, Hotelpackage\}$, from which we obtain the expected conclusion $K \cup T \mid\sim^{PQCL} Hotelpackage$.

5 Normalization Form

In this section, we show that any set of preferences, can equivalently be transformed into a set of normal form preferences, which are simply basic choice formulas.

We need to introduce the notion of equivalence between two formulas in QCL_{PS}. It is given by the following definition:

Definition 6. *Two QCL_{PS} formulas ϕ_1 and ϕ_2 are said to be equivalent, denoted simply by $\phi_1 \equiv \phi_2$, if:*

- *For all interpretation I, and integer k we have $I \mid\sim_k^{PQCL} \phi_1$ iff $I \mid\sim_k^{PQCL} \phi_2$.*
- *opt(ϕ_1) = opt(ϕ_2).*

The following introduces a normal form function, which associates with each general choice formula, its corresponding basic choice formula. This normal form function, denoted by $\mathcal{N}$, will allow us to transform any set of preferences into a set of basic choice formulas. This is very important from computation point of view, since it allow us to reuse various non-monotonic approaches such as possibilistic logic [10] or compilation of stratified knowledge bases [16].

Let ϕ be a formula in QCL_{PS}, let $\phi_1 = a_1 \times a_2 \times \ldots \times a_n$, and $\phi_2 = b_1 \times b_2 \times \ldots \times b_m$ be two basic choice formulas. The idea in defining the normal form of ϕ, denoted by $\mathcal{N}(\phi)$, is very simple. If ϕ is of the forme $\phi_1 \times \phi_2$ (resp. $\neg\phi_1$, $\phi_1 \wedge \phi_2$, $\phi_1 \vee \phi_2$) then $\mathcal{N}(\phi)$ is simply the result of applying $\times$ (resp. $\neg$, $\wedge$, $\vee$) to the normal form of its components. Namely:

1. $\mathcal{N}(\neg\phi_1) \equiv \mathcal{N}(\neg\mathcal{N}(\phi_1))$.
2. $\mathcal{N}(\phi_1 \wedge \phi_2) \equiv \mathcal{N}(\mathcal{N}(\phi_1) \wedge \mathcal{N}(\phi_2))$.
3. $\mathcal{N}(\phi_1 \vee \phi_2) \equiv \mathcal{N}(\mathcal{N}(\phi_1) \vee \mathcal{N}(\phi_2))$.

4. $\mathcal{N}(\phi_1 \times \phi_2) \equiv \mathcal{N}(\mathcal{N}(\phi_1) \times \mathcal{N}(\phi_2))$.
   If ϕ is already a basic choice formula, then $\mathcal{N}(\phi)$ is always equal to ϕ, namely
5. $\forall \phi_1 \in BCF_{PS}, \mathcal{N}(\phi_1) = \phi_1$.
   Hence, it only remains to define $\mathcal{N}(\phi_1 \wedge \phi_2)$ (resp. $\mathcal{N}(\phi_1 \vee \phi_2)$, $\mathcal{N}(\neg\phi_1)$) where ϕ_1 and ϕ_2 are normal form. This is given by items 6, 7, 8. Namely:
6. $\mathcal{N}(\phi_1 \wedge \phi_2) \equiv \ c_{11} \times \ldots \times \ c_{1m} \times \ c_{21} \times \ldots \times \ c_{2m} \times \ \ c_{n1} \times \ldots \times \ c_{nm}$, with $c_{ij} = \ a_i \wedge \ b_j$.
7. $\mathcal{N}(\phi_1 \vee \phi_2) \equiv \ d_{11} \times \ldots \times \ d_{1m} \times \ d_{21} \times \ldots \times \ d_{2m} \times \ \ d_{n1} \times \ldots \times \ d_{nm}$, with $d_{ij} = \ a_i \vee \ b_j$.
8. $\mathcal{N}(\neg\phi_1) \equiv \neg a_1 \times \neg a_2 \times ... \times \neg a_n$.

Property 6 confirms the meaning of prioritized conjunction. Indeed, assume $a_1 \times a_2 \times \ldots \times a_n$ denotes a preference of a user A, and $b_1 \times b_2 \times \ldots \times b_m$ denotes a preference of B. Applying, prioritized conjunction allows to select solutions that privileges A. For instance, $a_1 b_m$ (which represents the best choice for A and the worst choice for B) is preferred to $a_2 b_1$ (which represents the best choice for B and the second choice for A).

Proposition 7. *Let K be a set of propositional formulas and T be a set of general choice formulas. Let T' be a set of basic choice preferences obtained from T by replacing each ϕ in T by $\mathcal{N}(\phi)$, then $\forall \phi$, $K \cup T \mid\!\sim^{PQCL} \phi$ iff $K \cup T' \mid\!\sim^{PQCL} \mathcal{N}(\phi)$.*

A sketch of proof is provided in the appendix.

As a consequence, for any formula in QCL_{PS}, we have two possibilities to implement it:

1. Using a relation of satisfaction on general choice formulas, as indicated in Definition 2, or
2. Normalize or generate a set of basic choice formulas from any set of preferences, we then apply the inference relation from BCF theories as indicated in Proposition 3,

Example 4. Let us illustrate Proposition 7 by the following example. Let $\phi = ((a \times b) \vee \neg c) \wedge (c \times b)$, and $I = \{\text{b,c}\}$. We use two different ways to give the satisfaction degree of this formula.

1. We normalize the set of preferences into a set of normal form preferences :
   The formula $((a \times b) \vee \neg c) \wedge (c \times b)$ is a general choice formula, thus using the normal form function, we have
   $\mathcal{N}(((\text{a} \times b) \vee \neg c) \wedge (c \times b)) \equiv \mathcal{N}(\ \mathcal{N}\ ((\text{a} \times b) \vee \neg c) \wedge \mathcal{N}\ (\text{c} \times b))$,
   $\equiv \mathcal{N}([(\text{a} \vee \neg c) \times (b \vee \neg c)] \wedge (c \times b))$,
   $\equiv \mathcal{N}([[\text{a} \vee \neg c) \wedge c] \times [(a \vee \neg c) \wedge b] \times \ [(\text{b} \vee \neg c) \wedge c] \times [(b \vee \neg c) \wedge b])$,
   $\equiv (\text{a} \wedge c) \times ((a \vee \neg c) \wedge b) \times (b \wedge c) \times ((b \vee \neg c) \wedge b)$
   The obtained formula is a basic choice formula, using Proposition 3, we have $I \not\models (a \wedge c)$ and $I \not\models ((a \vee \neg c) \wedge b)$ but $I \models (b \wedge c)$, thus $I \mid\!\sim_3^{PQCL} \phi$.
2. Now, we use directly Definition 2 :
   The formula ϕ is of the form $(\phi_1 \wedge \phi_2)$ with $\phi_1 = (a \times b) \vee \neg c$ and $\phi_2 = c \times b$. We have $I \mid\!\sim_{j=1}^{PQCL} \phi_2$. The formula ϕ_1 is of the form $\phi' \vee \phi''$ with $\phi' = a \times b$

and $\phi'' = \neg c$. So we have $I \mid\sim^{PQCL}_{i'=2} a \times b$ and $I \not\models \neg c$, then $I \mid\sim^{PQCL}_{i} \phi_1$ and i = (i'-1)×$opt(\phi'')$ + 1 = (2-1)× 1 + 1 = 2. Lastly, we apply item (5) of Definition 2 we obtain, $I \mid\sim^{PQCL}_{k} \phi$ and k = (i-1)×opt(ϕ_2) + j = 3. Hence, we get the same result.

6 Conclusions

The problem of representing preferences has drawn attention from Artificial Intelligence researchers. The paper on preference logic [17] addresses the issue of capturing the common-sense meaning of preference through appropriate axiomatizations. The papers on preference reasoning [11,12,6] attempt to develop practical mechanisms for making inference about preferences and making decisions. A principal concept there is Ceteris Paribus preference: preferring one outcome to another, everything else being equal. The work on prioritized logic programming and non-monotonic reasoning [3,14,15] has potential applications to databases. CP-nets [6,8] use Bayesian-like structure to represent preferences under again Ceteris Paribus principle. However, in the majority of theses works, the negation in the context of representing preferences is not discussed, and few of them integrated prioritized preferences.

In this paper, a new logic for representing prioritized preferences has been proposed. This logic is characterized by new definitions of negation, conjunction and disjunction that are useful for aggregating preferences of users having different priority levels and overcome the QCL limitations. It generalizes the way the inference is done by presenting an inference framework based on normal form function or directly by using the satisfaction relation.

A future work is to apply our methods to alarms filtering. Indeed, existing alerts correlations and alerts filtering do not use take into account administrator's preferences. Our PQCL will be basically use to compactly represent administrator's preferences, and will be use to rank-order alerts to be presented to netwok administrator.

Acknowledgments

This work is supported by the SETIN06 project PLACID (Probabilistic graphical models and Logics for Alarm Correlation in Intrusion Detection).

References

1. Brewka, G., Benferhat, S., Le Berre, D.: Qualitative choice logic. Artif. Intell. 157(1-2), 203–237 (2004)
2. Brewka, G.: Logic Programming with Ordered Disjunction. In: Proc. AAAI-02, Edmonton, canada, pp. 100–105 (2002)
3. Brewka, G., Eiter, T.: Preferred answer sets for extended logic programs. Artif. Intell. 109(1-2), 297–356 (1999)

4. Benferhat, S., Le Berre, D., Sedki, K.: An Alternative Inference for Qualitative Choice Logic. In: Proc, ECAI'06, pp. 743–744 (2006)
5. Minker, J.: Logic-based Artificial Intelligence. Kluwer Academics Publisher, Boston (2000)
6. Boutilier, C., Brafman, R.I., Hoos, H.H., Poole, D.: Reasoning with conditional ceteris paribus preference statements. In: Proc. UAI-99, Stockhilm, Sweden, pp. 71–80 (1999)
7. Boutilier, C.: Towards a Logic for Qualitative Decision Theory. In: Proc. KR-94, pp. 75–86 (1994)
8. Domshlak, C., Brafman, R.I.: CP-nets - Reasoning and Consistency Testing. In: Proceedings of KR'02, pp. 121–132. Morgan Kaufmann, San Francisco (2002)
9. Lang, J., Der Torre, V., Weydert, E.: Utilitarian Desires. Autonomous Agents and Multi-Agent Systems 5(3), 329–363 (2002)
10. Dubois, D., Lang, J., Prade, H.: Possibilistic Logic. Handbook of Logic in Artificial Intelligence and Logic Programming 3, 439–513 (1994)
11. Wellman, M.P., Doyle, J.: Preferential semantics for goals. In: Proceedings of National Conference on artificial Intelligence, Anaheim, pp. 698–703 (1991)
12. Tan, S.W., Pearl, J.: Qualitative Decision Theory. In: Proc. AAAI-94, Seattle, pp. 928–933 (1994)
13. Tan, S.W., Pearl, J.: Specification and evaluation of preferences under uncertainty. In: KR'99, bonn, pp. 530–539 (1994)
14. Delgrande, J.P., Schaub, T., Tompits, H.: Logic programs with compiled preferences. In: Procedings of the European Conference an Artificial Intelligence, pp. 464–468 (2000)
15. Sakama, C., Inoue, K.: Prioritized logic programming and its application to commonsense reasoning. Artif. Intell 123, 185–222 (2000)
16. Marquis, S.C., Marquis, P.: On Stratified Belief Base Compilation. Annals of Mathematics and Artificial Intelligence 42(4), 399–442 (2004)
17. Wright, V.: The Logic of Preference. Edinburgh University Press, Edinburgh (1963)

Appendix. Proof of Proposition 7

For sake of space, we can only give the sketch of proof of the conjunction and disjunction. Let $\phi_1 = a_1 \times a_2 \times \ldots \times a_n$, and $\phi_2 = b_1 \times b_2 \times \ldots \times b_m$.

1. Let us give the proof of $I \mid\sim_k^{PQCL} (\phi_1 \wedge \phi_2)$ iff $I \models_k \mathcal{N}(\phi_1 \wedge \phi_2)$.
   Recall that : $\mathcal{N}(\phi_1 \wedge \phi_2) \equiv\ c_{11} \times \ldots \times c_{1m} \times c_{21} \times \ldots \times c_{2m} \times \ldots \times c_{n1} \times \ldots \times c_{nm}$, with $c_{ij} = a_i \wedge b_j$. Let us consider different cases of satisfaction of $a_i's$ and $b_i's$ by the interpretation I.
   - Suppose that there exists i > 0 and j > 0 such that $I \models_i a_1 \times \ldots \times a_n$ and $I \models_j b_1 \times \ldots \times b_m$, this also means that $I \models \neg a_1 \wedge ... \wedge \neg a_{i-1} \wedge a_i$ and $I \models \neg b_1 \wedge ... \wedge \neg b_{j-1} \wedge b_j$.

     This means that I falsifies $\{ c_{11}, \ldots, c_{1m}, c_{21}, \ldots, c_{2m}, \ldots, c_{i1}, \ldots, c_{i(j-1)}\}$, but I satisfies $c_{ij} (= a_i \wedge b_j)$. We have (i-1)×m + j -1 items which are not satisfied before satisfying $(a_i \wedge b_j)$. So, this means that $I \models_k c_{11} \times \ldots \times c_{1m} \times c_{21} \times \ldots \times c_{2m} \times c_{n1} \times \ldots \times c_{nm}$, namely $I \models_k \phi_1 \wedge \phi_2$ and k = (i-1)×m + j.

     Hence using item (5) of Definition 2, we can check that we also have $K \cup T \mid\sim_k^{PQCL} \phi_1 \wedge \phi_2$ and k= (i-1)×opt(ϕ_2) + j.

– There is no i such that $I \models_i \ a_1 \times \ldots \times a_n$ or there is no j such that $I \models_j \ b_1 \times \ldots \times b_m$. This means that I either falsifies all $a_i's$, or I falsifies all $b_j's$. Thus I falsifies c_{ij} ($= a_i \wedge b_j$), namely there is no k such that $I \models_k (\phi_1 \wedge \phi_2)$. Hence $K \cup T \not\sim_k^{PQCL} \phi_1 \wedge \phi_2$.

2. Let us give the proof of $I \sim_k^{PQCL} (\phi_1 \vee \phi_2)$ iff $I \models_k \mathcal{N}(\phi_1 \vee \phi_2)$.
Recall that : $\mathcal{N}(\phi_1 \vee \phi_2) \equiv \ d_{11} \times \ldots \times d_{1m} \times d_{21} \times \ldots \times d_{2m} \times \ldots \times d_{n1} \times \ldots \times d_{nm}$, with $d_{ij} = a_i \vee b_j$. Let us consider different cases of satisfaction of $a_i's$ and $b_i's$ by the interpretation I.
   – Suppose that there exists i > 0 or j > 0 such that $I \models_i \ a_1 \times \ldots \times a_n$ and $I \models_j \ b_1 \times \ldots \times b_m$. This means that there exists k > 0 such that $I \models_k \ d_k$. We can distinguish different cases :
      - If i = 1 or j = 1, then I satisfies $\{ d_{11}, \ d_{12}, \ldots, \ d_{1(j-1)}, \ d_{1j}, \ldots, \ d_{1m}, \ d_{21}, \ldots \ d_{i1}\}$ but I falsifies the rest of items. In this case, the first satisfied item is d_{11}, thus $I \models_1 \ d_{11} \times \ldots \times d_{1m} \times d_{21} \times \ldots \times d_{2m} \times \ldots \times d_{n1} \times \ldots \times d_{nm}$. Hence $I \models_1 (\phi_1 \vee \phi_2)$.
      Using item (4)-a of Definition 2, we can check that we also have $K \cup T \sim_k^{PQCL} \phi_1 \vee \phi_2$ and k =1.
      - If i > 1 or j > 0, then I falsifies $\{ \ d_{11}, \ d_{12}, \ldots, \ d_{1(j-1)}, \ d_{21}, \ldots, \ d_{2(j-1)}, \ldots \ d_{i1}, \ldots, \ d_{i(j-1)}\}$ but I satisfies $\{ \ d_{1j}, \ldots, \ d_{2j}, \ldots, \ d_{ij}\}$.
      In this case, we have (j-1) not satisfied items before the first satisfied item d_{1j}, thus $I \models_j \ d_{11} \times \ldots \times d_{1m} \times d_{21} \times \ldots \times d_{2m} \times \ldots \times d_{n1} \times \ldots \times d_{nm}$. Hence $I \models_j (\phi_1 \vee \phi_2)$.
      Using item (4)-c of Definition 2, we have $K \cup T \sim_k^{PQCL} \phi_1 \vee \phi_2$ and k =j.
   – There is no i such that $I \models_i \ a_1 \times \ldots \times a_n$ and there is j such that $I \models_j \ b_1 \times \ldots \times b_m$. This means that $I \models \neg a_1 \wedge \ldots \wedge \neg a_{i-1} \wedge \neg a_i \vee \ldots \vee \neg a_n$ and $\exists$ j > 0 such that $I \models \neg b_1 \wedge \ldots \wedge \neg b_{j-1} \wedge b_j$.
      This also means that I falsifies $\{ \ d_{11}, \ldots, \ d_{1(j-1)}, \ldots, \ d_{21}, \ldots, \ d_{2(j-1)} \ \ldots, \ d_{(i-1)(j-1)}\}$, but I satisfies the items $\{d_{1j}, \ d_{2j}, \ldots, \ d_{(i-1)j}, \ d_{ij}\}$. So, d_{1j} is the first satisfied item and all the items before d_{1j} are not satisfied, namely we have at least (j-1) not satisfied items before the first satisfied item $(a_1 \vee b_j)$, this means that $I \models_k \ d_{11} \times \ldots \times d_{1m} \times d_{21} \times \ldots \times d_{2m} \times \ldots \times d_{n1} \times \ldots \times d_{nm}$, hence $I \models_j (\phi_1 \vee \phi_2)$.
      Using item (4)-c of Definition 2, we have also $K \cup T \sim_k^{PQCL} (\phi_1 \vee \phi_2)$, and k = j.
   – There is i such that $I \models_i \ a_1 \times \ldots \times a_n$ and there is no j such that $I \models_j \ b_1 \times \ldots \times b_m$. This means that there $\exists$ i > 0 such that $I \models \neg a_1$ *wedge*... $\wedge \neg a_{i-1} \wedge a_i$ and $\nexists$ j such that $I \models \neg b_1 \wedge \ldots \wedge \neg b_{j-1} \wedge \neg b_j$.
      This means that I falsifies $\{ \ d_{11}, \ldots, \ d_{1(j-1)}, d_{1j} \ldots, \ d_{21}, \ldots, \ d_{2(j-1)}, \ d_{2j}$,
      ..., $d_{(i-1)(j-1)}\}$, but I satisfies the items $\{d_{i1}, \ \ldots, \ d_{(i)(j-1)}, \ d_{ij}\}$. So, d_{i1} is the first satisfied item and all the items before d_{i1} $(= a_i \vee b_1)$ are not satisfied, namely we have at least (i-1)×m not satisfied items, this means that $I \models_k \ d_{11} \times \ldots \times d_{1m} \times d_{21} \times \ldots \times d_{2m} \times \ldots \times d_{n1} \times \ldots \times d_{nm}$, so $I \models_k (\phi_1 \vee \phi_2)$, and k = (i-1)×m +1.

Hence using item (4)-b of Definition 2, we have also $K \cup T \mid\sim_k^{PQCL} (\phi_1 \vee \phi_2)$, and k = (i-1)×opt($\phi_2$) +1.

– There is no i such that $I \models_i \ a_1 \times \ldots \times \ a_n$ and there is no j such that $I \models_j \ b_1 \times \ldots \times \ b_m$. This means that I either falsifies all $a_i's$, I falsifies all $b_j's$. Hence I falsifies all d_{ij} (= $a_i \vee \ b_j$).

Hence, there is no k such that $I \models_k (\phi_1 \vee \phi_2)$, so $K \cup T \not\mid\sim^{PQCL} (\phi_1 \vee \phi_2)$.

An Abstract Theory of Argumentation That Accommodates Defeasible Reasoning About Preferences

Sanjay Modgil

Advanced Computation Lab, Cancer Research UK, London

Abstract. Dung's abstract theory of argumentation has become established as a general framework for non-monotonic reasoning, and, more generally, reasoning in the presence of conflict. In this paper we extend Dung's theory so that an argumentation framework distinguishes between: 1) attack relations modelling different notions of conflict; 2) arguments that themselves claim preferences, and so determine defeats, between other conflicting arguments. We then define the acceptability of arguments under Dung's extensional semantics. We claim that our work provides a general unifying framework for logic based systems that facilitate defeasible reasoning about preferences. This is illustrated by formalising argument based logic programming with defeasible priorities in our framework.

1 Introduction

A Dung argumentation framework [7] consists of a set of arguments $Args$ and a binary conflict based relation $\mathcal{R}$ on $Args$. A 'calculus of opposition' is then applied to the framework to evaluate the winning (justified) arguments under different extensional semantics. The underlying logic, and definition of the logic's constructed arguments $Args$ and relation $\mathcal{R}$, is left unspecified, thus enabling instantiation of a framework by various logical formalisms. Dung's seminal theory has thus become established as a general framework for non-monotonic reasoning, and, more generally, reasoning in the presence of conflict. A theory's inferences can be defined in terms of the claims of the acceptable arguments constructed from the theory (an argument essentially being a proof of a candidate inference - the argument's claim - in the underlying logic). Indeed, many of the major species of logic programming and non-monotonic logics (e.g. default, autoepistemic, non-monotonic modal logics and certain instances of circumscription) turn out to be special forms of Dung's theory [7,5].

Dung's abstract framework has been refined (e.g., [1] [4]) to explicitly model the role of preferences. The relation $\mathcal{R}$ can then denote either *attack* or *defeat* between arguments, where defeat represents a successful attack by additionally accounting for the relative strengths of (preferences between) attacking arguments. However, preference information is assumed pre-specified and external to the underlying logical formalism. This contrasts with the way people normally argue and reason *about*, as well as *with*, uncertain, defeasible and possibly conflicting preference information. This has led to works extending the underlying object level logical languages with rules for deriving priorities amongst rules; for example, in default logic [6] and logic programming formalisms [11]. One can then construct 'priority arguments' whose claims determine preferences between other mutually attacking arguments to determine the direction of

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 648–659, 2007.

the successful attacks (defeats). Arguments claiming conflicting priorities may be constructed and preferences between these can be established on the basis of other priority arguments. However, these works are restricted to basing argument strength on the priorities of their constituent rules. More recent work [9,10] organises Dung frameworks into a hierarchy so that given mutually attacking arguments in a level n framework, one can reason about the strengths of, and relative preferences between these arguments (rather than their constituent rules) in a first order logic instantiating a level $n+1$ framework, and thus determine the direction of defeat between the n framework arguments. This allows for argument strength based on a range of criteria, including criteria that relate to the argument as a whole, such as the value promoted by the argument [4]. However, a limitation is that the separation between object level n and meta-level $n+1$ reasoning means that argumentation at level n cannot, when appropriate, affect the outcome of argumentation at level $n+1$.

In this paper we extend Dung's framework to include arguments that *claim preferences between other arguments* and so determine whether attacks succeed as defeats. The extended framework also distinguishes between different types of conflict (attack) that have been formalised in the literature. These differences manifest in terms of how defeat is then determined. We then define evaluation of the justified arguments of an extended framework under Dung's extensional semantics. We aim at an abstract unifying theory in which approaches of the type described in the preceding paragraph can now be formalised and extended. This is illustrated by an example taken from [10], in which argumentation about values and value orderings can now be formalised in a single extended framework. We also formalise [11]'s argument based logic programming with defeasible priorities in our extended framework. In contrast with [11], we can then evaluate the justified arguments under *all* of Dung's semantics. We also claim our work will facilitate future development of argumentation systems formalising argumentation about preferences. We comment further on future work in the concluding section.

2 Extended Argumentation Frameworks

Given a Dung argumentation framework $(Args, \mathcal{R})$ where $\mathcal{R} \subseteq (Args \times Args)$ is the attack relation, we summarise how existing argumentation based formalisms evaluate the defeat relation given a preference relation on arguments. If A and B symmetrically attack ((A, B), (B, A) $\in \mathcal{R}$), then A (B) defeats B (A) if B (A) is not preferred to A (B). We call such attacks *preference dependent*. An example of a symmetric attack is when A and B claim logically contradictory conclusions. Examples of preference dependent asymmetric attacks - $(A, B) \in \mathcal{R}$ and $(B, A) \notin \mathcal{R}$ - also occur. A does not defeat B if B is preferred to A. This may lead to both A **and** B being justified, which is appropriate only if A and B do not logically contradict. For example, in value based argumentation over action [3], B may justify an action, and A claims the action has an undesirable side-effect. Preferring B to A may then result in both arguments being justified; the action is chosen while acknowledging it's undesirable side-effect. Asymmetric attacks may also be *preference independent* (*pi*) in that A defeats B irrespective of any relative preference. For example, in logic programming systems (e.g.[11]) in which A proves (claims) what was assumed non-provable (through negation as failure) by B.

Suppose now that A preference dependent (*pd*) attacks B, and there is an argument C claiming a preference for B over A. Hence A does not successfully attack (defeat) B. Intuitively, C is an argument for B's repulsion of, or defence against, A's attack on B. We can say that C *defence attacks* A***'s attack on*** B. If in addition, B *pd* attacks A, then B now defeats A unless there is some C' claiming a preference for A over B and so *defence* (*d*) attacking B's attack on A. C and C' claim contradictory preferences and so *pd* attack each other. These *pd* attacks can themselves be subject to *d* attacks in order to determine the defeat relation between C and C' and so A and B. We now formally define the elements of an *Extended Argumentation Framework*.

Definition 1. *An* Extended Argumentation Framework *(*EAF*) is a tuple* $(Args, \mathcal{R}_{pd}, \mathcal{R}_{pi}, \mathcal{R}_d)$ *such that* $Args$ *is a set of arguments, and:*

- $\mathcal{R}_{pi} \subseteq Args \times Args$
- $\mathcal{R}_{pd} \subseteq Args \times Args$
- $\mathcal{R}_d \subseteq (Args \times \mathcal{R}_{pd})$
- *If* $(A, (B, C)), (A', (C, B)) \in \mathcal{R}_d$ *then* $(A, A'), (A', A) \in \mathcal{R}_{pd}$

Notation 1. From hereon, if every *pd* attack in $\mathcal{R}_{pd}$ is symmetric, then we refer to the *EAF* as a symmetric *EAF* (*sEAF*). Also:

- $A \rightharpoonup B$ denotes $(A, B) \in \mathcal{R}_{pd}$. If in addition $(B, A) \in \mathcal{R}_{pd}$, we may write $A \rightleftharpoons B$.
- $A \hookrightarrow B$ denotes $(A, B) \in \mathcal{R}_{pi}$
- $C \twoheadrightarrow (A \rightharpoonup B)$ denotes $(C,(A, B)) \in \mathcal{R}_d$.

From hereon definitions are assumed relative to an *EAF* $(Args, \mathcal{R}_{pd}, \mathcal{R}_{pi}, \mathcal{R}_d)$, where arguments $A, B, \ldots$ are assumed to be in $Args$, and S is a subset of $Args$. We now formally define two defeat relations that are parameterised w.r.t. some set S of arguments. This accounts for a *pd* attack's success being relative to preference arguments in S, rather than relative to some externally given preference ordering.

Definition 2

- *A S-1-defeats B iff* $(A, B) \in \mathcal{R}_{pi}$*; or* $(A, B) \in \mathcal{R}_{pd}$ *and* $\neg\exists C \in S$ *s.t.* $(C,(A, B)) \in \mathcal{R}_d$
- *A S-2-defeats B iff* $(A, B) \in \mathcal{R}_{pi}$*; or* $(A, B) \in \mathcal{R}_{pd}$ *and* $(B, A) \notin \mathcal{R}_{pi}$ *and* $\neg\exists C \in S$ *s.t.* $(C,(A, B)) \in \mathcal{R}_d$
- *A* strictly *S-1(2)-defeats B iff A S-1(2)-defeats B and B does not S-1(2)-defeat A.*

S-2-defeat is deployed in works such as [11] in which *pi* attacks over-ride contrary *pd* attacks. From hereon we simply write 'S-defeat', and only distinguish between S-1 and S-2 when relevant. Also, we may write $A \rightarrow^S B$ to denote that A S-defeats B.

Example 1. Let Δ be the *EAF*:

$$A \rightleftharpoons B, C \twoheadrightarrow (A \rightharpoonup B)$$

A and B S-defeat each other for $S = \emptyset$, $\{A\}$ and $\{B\}$. B $\{C\}$-defeats A but A does not $\{C\}$-defeat B (B strictly $\{C\}$-defeats A). If Δ also contained the *pi* attack $A \hookrightarrow B$, then for any $S \subseteq Args$, B S-1-defeats A but does not S-2-defeat A.

We now define the notion of a conflict free set $S \subseteq Args$.

Definition 3. *S is conflict free iff $\forall A, B \in S$:*

1. *$(A, B) \notin \mathcal{R}_{pi}$; and*
2. *if $(A, B) \in \mathcal{R}_{pd}$ then $(B, A) \notin \mathcal{R}_{pd}$, and $\exists C \in S$ s.t. $(C,(A, B)) \in \mathcal{R}_d$.*

Suppose $S = \{A, B, C\}$, where $C \twoheadrightarrow (A \rightharpoonup B)$ and it is not the case that $B \rightharpoonup A$. Hence, A and B do not S-defeat each other and S is conflict free. Note that $S' = \{A, B\}$ is not conflict free. Only for symmetric EAFs in which every *pd* attack is symmetric, can one show that every subset of a conflict free set is conflict free.

3 Defining Acceptability Semantics

Given a Dung framework $(Args, \mathcal{R})$, a **single** argument A is defined as acceptable w.r.t. some $S \subseteq Args$, if for every B such that $(B, A) \in \mathcal{R}$, there exists a $C \in S$ such that $(C, B) \in \mathcal{R}$. Intuitively, C 'reinstates' A. Dung then defines the acceptability of a **set** of arguments under different extensional semantics. The definition is given here, where $S \subseteq Args$ is conflict free if no two arguments in S are related by $\mathcal{R}$.

Definition 4. *Let $S \subseteq Args$ be a conflict free set. Then:*

- *S is an admissible extension iff each argument in S is acceptable w.r.t. S*
- *S is a preferred extension iff S is a set inclusion maximal admissible extension*
- *S is a complete extension iff each argument which is acceptable w.r.t. S is in S*
- *S is a stable extension iff $\forall B \notin S$, $\exists A \in S$ such that $(A, B) \in \mathcal{R}$*

We now motivate definition of the acceptability of an argument w.r.t. a set S for an *EAF*, so that we can then apply the above defined extensional semantics. Consider example 1. Is A acceptable w.r.t. $S = \{A\}$? (which amounts to asking whether $\{A\}$ is admissible). We have that $B \rightarrow^S A$. The **only** argument that can reinstate A is A itself, via the defeat $A \rightarrow^S B$ that is based on A successfully *pd* attacking B (i.e., $\{A\}$'s admissibility is contingent on B **not** being preferred to A). However, the success of A's *pd* attack on B is challenged by the *d* attack from the argument C (expressing that B **is** preferred to A). Hence, we need to ensure that C is S-defeated by some argument in S that effectively 'reinstates' the S-defeat $A \rightarrow^S B$. There is no such argument and so A is not acceptable w.r.t. S. Intuitively, since C is not attacked we will always conclude a preference for B over A. This precludes A being acceptable w.r.t. any S since this requires that B not be preferred to A. The above suggests the following definition of the 'local' acceptability of an S-defeat w.r.t. a set S:

Definition 5. *Let $(Args, \mathcal{R}_{pd}, \mathcal{R}_{pi}, \mathcal{R}_d)$ be an* EAF. *Then $C \rightarrow^S B$ is* locally acceptable *w.r.t. $S \subseteq Args$ iff:*

1. *$(C, B) \in \mathcal{R}_{pi}$; or*
2. *if $(C, B) \in \mathcal{R}_{pd}$, then $\forall B' \in Args$ s.t. $(B',(C, B)) \in \mathcal{R}_d$, there $\exists C' \in S$ s.t. $C' \rightarrow^S B'$*

However, the above definition does not suffice for *EAF*s that contain asymmetric *pd* attacks. In figure 1-a), the acceptability of $A1$ w.r.t. S (and so the admissibility of S given that $A2$ is not attacked and so must be acceptable) is under consideration. $B1 \rightarrow^S$

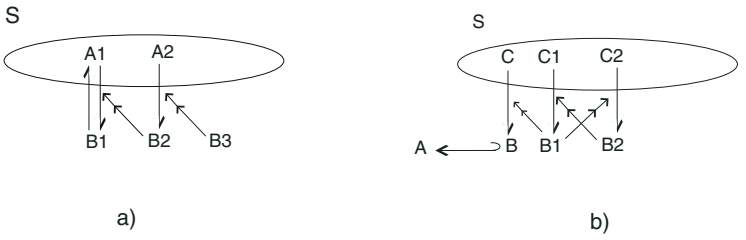


Fig. 1. Checking the acceptability of arguments $A1$ and A w.r.t. S in a) and b) respectively

$A1$ and $A1 \rightarrow^S B1$, and by definition 5, $A1 \rightarrow^S B1$ is acceptable w.r.t. S since $A2 \rightarrow^S B2$. However, an admissible extension corresponds to a defendable position in which all arguments can coherently said to be 'winning', and $\{A1, A2\}$ does not represent such a position. Intuitively, $B3$, expressing that $B2$ is preferred to $A2$, is not attacked and so must be a winning argument. This means that $A2$ cannot successfully *pd* attack $B2$, and since this *pd* attack is asymmetric, $A2$ **and** $B2$ (expressing a preference for $B1$ over $A1$) must be winning. Therefore, $A1$ cannot be a member of an admissible extension since its membership is contingent on $B1$ not being preferred to $A1$. Hence, we should therefore check the 'global' acceptability of $A1 \rightarrow^S B1$, in the sense that $A2 \rightarrow^S B2$ should itself be acceptable w.r.t. S; it is not, since no argument in S S-defeats $B3$.

Definition 6. *Given (Args, $\mathcal{R}_{pd}$, $\mathcal{R}_{pi}$, $\mathcal{R}_d$), $C \rightarrow^S B$ is* globally acceptable *w.r.t. $S \subseteq Args$ iff there exists a set of S defeats $\mathcal{D}_S = \{X_1 \rightarrow^S Y_1, \ldots, X_n \rightarrow^S Y_n\}$, such that:*

1. $C \rightarrow^S B \in \mathcal{D}_S$
2. *for* $i = 1 \ldots n$, $X_i \in S$
3. $\forall X \rightarrow^S Y \in \mathcal{D}_S$ *s.t.* $(X, Y) \in \mathcal{R}_{pd}$, $\forall Y'$ *s.t.* $(Y', (X, Y)) \in \mathcal{R}_d$, *there* $\exists X' \rightarrow^S Y' \in \mathcal{D}_S$

in which case we say that $\mathcal{D}_S$ is a reinstatement set *for $C \rightarrow^S B$.*

We can now define two notions of acceptability:

Definition 7. *A is* locally, *respectively* globally, *acceptable w.r.t. S, iff:*
$\forall B$ s.t. $B \rightarrow^S A$, $\exists C \in S$ s.t. $C \rightarrow^S B$ and $C \rightarrow^S B$ is locally, *respectively* globally, *acceptable w.r.t. S.*

In figure 1-b), A is locally and globally acceptable w.r.t. S. In the latter case there is a reinstatement set for $C \rightarrow^S B$: $\{C \rightarrow^S B, C1 \rightarrow^S B1, C2 \rightarrow^S B2\}$. Note that if in addition there was an argument $B3$ such that $B3 \twoheadrightarrow (C2 \rightharpoonup B2)$ and no argument in S that S defeats $B3$, then no reinstatement set for $C \rightarrow^S B$ would exist. A would be locally, but not globally, acceptable w.r.t. S.

Extensional semantics for *EAF*s are now given by definition 4, assuming either local or global acceptability, conflict free defined as in definition 3, and for the stable

semantics, 'A S-defeats B' replaces '$(A, B) \in \mathcal{R}$'. For the complete, preferred and stable semantics, an argument is said to be sceptically justified if it belongs to all extensions, and credulously justified if it belongs to at least one extension. From hereon, we will assume the extensional semantics defined in terms of global acceptability, and simply write 'acceptable' rather than 'globally acceptable'. However, note the following[1]:

Proposition 1. *Let Δ be a symmetric* EAF. *S is an admissible extension of Δ as defined by local acceptability iff S is an admissible extension of Δ as defined by global acceptability.*

We now state that Dung's fundamental lemma [7] and the theorem that follows from this lemma also hold for *EAF*s:

Lemma 1. *Let S be an admissible extension of an* EAF*, and let A and B be arguments that are acceptable w.r.t S. Then:*

- *$S' = S \cup \{A\}$ is admissible*
- *B is acceptable w.r.t. S'*

Theorem 1. *Let Δ be an* EAF.

1. *The set of all admissible extensions of Δ form a complete partial order w.r.t. set inclusion*
2. *For each admissible S there exists a preferred extension S' of Δ such that $S \subseteq S'$*

Proof of lemma 1 shows that for the lemma not to hold, A would have to d attack some $(C, B) \in \mathcal{R}_{pd}$ where $C \rightarrow^S B$ is in a reinstatement set that licenses the acceptability of A (A') w.r.t. S. But then this would contradict A being acceptable w.r.t. S. Given theorem 1 and the fact that $\emptyset$ is an admissible extension of every *EAF*, we can state that every *EAF* possesses at least one preferred extension. Theorem 1 also implies that it is sufficient to show an admissible extension containing A to determine whether A is credulously justified under the preferred semantics. Note also, that it is straightforward to show that every stable extension of an *EAF* is a preferred extension but not vice versa.

In [7], the characteristic function F of a Dung framework $(Args, \mathcal{R})$ is defined, such that for any $S \subseteq Args$, $F(S) = \{A | A$ is acceptable w.r.t. $S\}$. If S is conflict free, and $S \subseteq F(S)$, respectively $S = F(S)$, then S is an admissible, respectively complete, extension. Dung then also defines the inherently sceptical *grounded* semantics, by defining the *grounded* extension as the least fixed point of F. Here, we similarly define the grounded extension of a symmetric *EAF*.

Definition 8. *Let Δ = ($Args$, $\mathcal{R}_{pd}$, $\mathcal{R}_{pi}$, $\mathcal{R}_d$) be a symmetric* EAF *and let 2^{ArgsC} be the set of all conflict free subsets of $Args$. Then the characteristic function F of Δ is defined as follows:*

- *$F : 2^{ArgsC} \mapsto 2^{Args}$*
- *$F(S) = \{A \in Args | A$ is acceptable w.r.t. $S\}$*

[1] Space limitations preclude inclusion of proofs here. However all proofs can be found in a technical report at http://acl.icnet.uk/ sm/TechnicalReport.pdf.

Proposition 2. *F is monotonic, i.e., if $S \subseteq S'$ then $F(S) \subseteq F(S')$.*

The monotonicity of F gives it a constructive flavour and guarantees the existence of a least fixed point. The following also holds:

Proposition 3. *Let $S \subseteq F(S)$. If S is conflict free then $F(S)$ is conflict free.*

Defining a sequence $\mathrm{F}^1 = F(\emptyset)$, $\mathrm{F}^{i+1} = F(\mathrm{F}^i)$, propositions 2 and 3 imply that $\mathrm{F}^{i+1} \supseteq \mathrm{F}^i$, where each F^j in the sequence is conflict free. Indeed, the sequence can be used to construct the least fixed point of a 'finitary' symmetric *EAF* in which for any argument A or *pd* attack (B, C), the set of arguments attacking A, respectively (B, C), is finite.

Proposition 4. *Let Δ be a symmetric* EAF *and let the following sequence be defined:*

$$F^1 = F(\emptyset),\ F^{i+1} = F(F^i)$$

If Δ is finitary then $\bigcup_{i=1}^{\infty}(F^i)$ is the least fixed point (grounded extension) of Δ.

For $\Delta 1$ in fig.2, $\mathrm{F}^1 = \{A, H\}$, $\mathrm{F}^2 = \{A, H, G\}$, $\mathrm{F}^3 = \{A, H, G, C\}$, $\mathrm{F}^4 = \{A, H, G, C, E\}$ where $F(\mathrm{F}^4) = \mathrm{F}^4$ is the grounded extension.

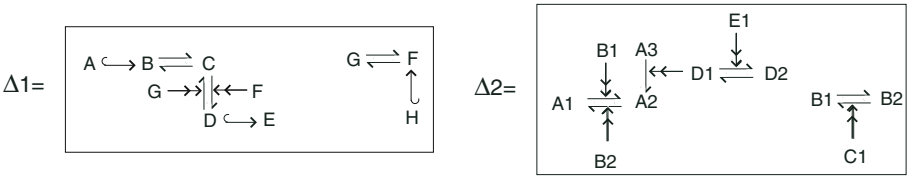


Fig. 2. *EAFs* $\Delta 1$ and $\Delta 2$

Notice that F applied to conflict free sets for *EAF*s containing asymmetric *pd* attacks is not monotonic. Consider $C \rightharpoonup B \hookrightarrow A$, $B \twoheadrightarrow (C \rightharpoonup B)$, where A is acceptable w.r.t. $\{C\}$ but not w.r.t. the conflict free superset $\{C, B\}$. This makes discussion of the grounded semantics for arbitrary *EAF*s more involved, and is beyond the scope of this paper[2]. Note also, that given the following notion of 'strict acceptability':

Definition 9. *A is strict-acceptable w.r.t. S iff $\forall B$ s.t. B S-defeats A, $\exists\, C \in S$ s.t. C* strictly *S-defeats B (where strict S defeat is defined as in definition 2).*

then lemmas 2 and 3 below imply that the grounded extension of a symmetric *EAF* can be obtained by a characteristic function defined in terms of strict-acceptability.

Lemma 2. *Let S be an admissible extension of a symmetric* EAF. *If A is strict-acceptable w.r.t. S then A is acceptable w.r.t. S.*

Lemma 3. *Let F^i be defined for a symmetric* EAF *as in proposition 4. Then $\forall A \in F^i$, if $\exists B$ s.t. B F^i-defeats A, then $\exists C \in F^i$ such that C strictly F^i-defeats B.*

[2] However, note that assuming the iteration of F in proposition 4 w.r.t. arbitrary *EAF*s, it can be shown that $\forall i, F^i \subseteq F^{i+1}$.

Proposition 5. *Let Δ be a symmetric* EAF*, F_{st} be defined (as in definition 8) on the basis of strict-acceptability, and $F^1_{st} = F_{st}(\emptyset)$, $F^{i+1}_{st} = F_{st}(F^i_{st})$. Then $\forall i$, $F^i_{st} = F^i$.*

Referring to $\Delta 1$ in figure 2, E is strict-acceptable w.r.t. F^3 since D F^3-defeats E, and since $G \in \mathrm{F}^3$, C strictly F^3-defeats D.

As discussed in the introduction, a number of works (e.g. [1] [4]) extend Dung argumentation frameworks with pre-defined preference orderings that enable definition of *defeat* given an *attack* relation. In [4], if A attacks B, then A defeats B only if the value promoted by B is not ranked higher than the value promoted by A according to some given pre-defined ordering on values. We conclude this section with an example that extends [4] to illustrate argumentation based reasoning about values and value orderings. The example is taken from [10] in which argumentation about preferences in a level n Dung framework determines the successful attacks (defeats) in a level $n-1$ framework. Here, the argumentation now takes place in a single *EAF*. Consider $\Delta 2$ in fig.2 in which $A1$ and $A2$ are arguments for the medical actions 'give aspirin' and 'give chlopidogrel' respectively. These arguments relate the current beliefs that warrant the actions bringing about states of affairs that realise a desired goal and so promote a value. We assume their construction in a BDI logic, as described in [3]. $A1$ and $A2$ both promote the value of *efficacy*. They symmetrically *pd* attack since they claim alternative actions for the goal of preventing blood clotting. In [10], first order argumentation systems are defined for reasoning about possibly conflicting valuations of arguments and value orderings. Argument $B1$ is based on clinical trial 1's conclusion that $A2$'s chlopidogrel is more efficacious than $A1$'s aspirin at preventing blood clotting. Hence $B1 \twoheadrightarrow (A1 \rightharpoonup A2)$. However, $B2$ is based on clinical trial 2's conclusion that the opposite is the case. Hence $B1 \rightleftharpoons B2$. At this stage neither $A1$ or $A2$ are sceptically justified under the preferred/complete/stable semantics. However, $C1$ is an argument claiming that trial 1 is preferred to trial 2 since the former uses a more statistically robust design. Now $A2$ and not $A1$ is sceptically justified. However, $A3$ promoting the value of *cost*, states that chlopidogrel is prohibitively expensive and so asymmetrically *pd* attacks $A2$. However, $D1 \twoheadrightarrow (A3 \rightharpoonup A2)$ where $D1$ is a value ordering ranking *efficacy* over *cost*. Hence, $A3$ does not defeat $A2$ and so $A2$ remains sceptically justified. However, $D2$ now ranks *cost* over *efficacy*. Now neither $A2$ or $A1$ are sceptically justified. Finally, $E1$ is a utilitarian argument stating that since financial resources are low, use of chlopidogrel will compromise treatment of other patients, and so one should preferentially rank *cost* over *efficacy* (such a trade of is often made in medical contexts). Hence, $A1$ is now sceptically justified; aspirin is now the preferred course of action.

4 Formalising Logic Programming with Defeasible Priorities in an EAF

In this section we show how an *EAF* can be instantiated by the arguments and their relations defined by a modified version of [11]'s *argument based logic programming with defeasible priorities* (ALP-DP). In ALP-DP , (S, D) is a theory where S is a set of strict rules of the form $s : L_0 \wedge \ldots \wedge L_m \rightarrow L_n$ and D a set of defeasible rules $r : L_0 \wedge \ldots \wedge L_j \wedge \sim L_k \wedge \ldots \wedge \sim L_m \Rightarrow L_n$. Each rule name r (s) is a first order

term. From hereon *head*(r) denotes the consequent L_n of the rule named r. Each L_i is a strong literal, i.e., an atomic first order formula, or such a formula preceded by strong negation $\neg$. Weak negation is denoted by $\sim$, so that each $\sim L_i$ is a weak literal read as "there is no evidence that L_i is the case". For any atom A, we say that A and $\neg A$ are the complement of each other. In the metalanguage, $\overline{L}$ denotes the complement of a literal L. As usual, a rule with variables is a scheme standing for all its ground instances. The language contains a two-place predicate symbol $\prec$ for expressing priorities on rule names. Any S is also assumed to contain the strict rules:

- $o1 : (x \prec y) \wedge (y \prec z) \rightarrow (x \prec z)$
- $o2 : (x \prec y) \wedge \neg(x \prec z) \rightarrow \neg(y \prec z)$
- $o3 : (y \prec z) \wedge \neg(x \prec z) \rightarrow \neg(x \prec y)$
- $o4 : (x \prec y) \rightarrow \neg(y \prec x)$

Definition 10. *An argument A based on the theory (S, D) is:*

1. *a finite sequence $[r_0, \ldots, r_n]$ of ground instances of rules such that:*
   - *for every i ($0 \leq i \leq n$), for every strong literal L_j in the antecedent of r_i there is a $k < i$ such that* head(r_k) = L_j. *If* head(r_n) = $x \prec y$ *then A is called a 'singleton priority argument'.*
   - *no distinct rules in the sequence have the same head;*

or:

2. *a finite sequence $[r_{0_1}, \ldots r_{n_1}, \ldots, r_{0_m}, \ldots r_{n_m}]$, such that for i=1 ... m, $[r_{0_i}, \ldots r_{n_i}]$ is a singleton priority argument. We say that A is a 'composite priority argument' that concludes the ordering* $\bigcup_{i=1}^{m}$ head(r_{n_i})

In ALP-DP, arguments are exclusively defined by 1). Preferences between arguments are then parameterised w.r.t. a set T, based on the ordering claimed by the *set* of singleton priority arguments in T. Here, we have additionally defined composite priority arguments in 2), so that an ordering, and hence a preference, is claimed by a single argument. This is the only modification we introduce to ALP-DP as defined in [11].

Example 2. Let $S = \{o1 \ldots o4\}$ and D be the set of rules:
r1 : $\Rightarrow a$, r2 : $\Rightarrow \neg a$, r3 : $a \Rightarrow b$, r4 : $\neg a \Rightarrow \neg b$,
r5 : $\Rightarrow r2 \prec r1$, r6 : $\Rightarrow r1 \prec r2$, r7 : $\Rightarrow r4 \prec r3$, r8 : $\Rightarrow r6 \prec r5$
Amongst the arguments that can be constructed are:
A1 = [r1], A2 = [r2], A3 = [r1, r3], A4 = [r2, r4], B1 = [r5], B2 = [r6], B3 = [r5, r7], B4 = [r6, r7], B5 = [r7], C1 = [r8]

The following definitions assume arguments are relative to a theory (S, D). [11] defines:

Definition 11. *For any arguments A, A' and literal L:*

- *A is strict iff it does not contain any defeasible rule; it is defeasible otherwise.*
- *A' is a sub-argument of A iff A' is a subsequence of A.*
- *L is a conclusion of A iff L is the head of some rule in A*
- *L is an assumption of A iff $\sim \overline{L}$ occurs in some antecedent of a rule in A.*
- *If T is a sequence of rules, then $A + T$ is the concatenation of A and T*

[11] motivates definition of attacks between arguments that account for the ways in which arguments can be extended with strict rules:

Definition 12. *$A1$ attacks $A2$ iff there are sequences S1 and S2 of strict rules such that $A1 + S1$ is an argument with conclusion L and*

1. *$A2 + S2$ is an argument with a conclusion $\overline{L}$, in which case $A1$ and $A2$ are said to symmetrically conclusion-conclusion attack on the the pair (L,$\overline{L}$); or*
2. *$A2$ is an argument with an assumption $\overline{L}$, in which case $A1$ is said to undercut $A2$*

In example 2, if we had the additional rules $s : a \rightarrow b$ and $r9 :\sim b \Rightarrow c$, then $A1 + [s]$ would undercut $A5$ = [r9]. Note that $B3$ = [r5, r7] and $B4$ = [r6, r7] conclusion-conclusion attack since [r5, r7] has the conclusion $r2 \prec r1$ and [r6, r7] + $[o4]$ has the conclusion $\neg(r2 \prec r1)$. Also, A3 and A4 conclusion-conclusion attack on the pairs $(a, \neg a)$ and $(b, \neg b)$. To determine a preference amongst conclusion-conclusion attacking arguments ALP-DP defines the sets of relevant rules to be compared:

Definition 13. *If $A+S$ is an argument with conclusion L, the defeasible rules $R_L(A+S)$ **relevant** to L are:*

1. *$\{r_d\}$ iff A includes defeasible rule r_d with head L*
2. *$R_{L_1}(A + S) \cup \ldots \cup R_{L_n}(A + S)$ iff A is defeasible and S includes a strict rule $s : L_1 \wedge \ldots \wedge L_n \rightarrow L$*

We define ALP-DP's ordering on these sets and hence preferences amongst arguments, on the basis of an ordering concluded by a single composite priority argument (rather than a set of singleton priority arguments as in ALP-DP):

Definition 14. *Let C be a priority argument concluding the ordering $\prec$. Let R and R' be sets of defeasible rules. Then $R' > R$ iff $\forall r' \in R'$, $\exists r \in R$ such that $r \prec r'$.*

Definition 15. *Let C be a priority argument concluding $\prec$. Let $(L_1, \overline{L_1}), \ldots, (L_n, \overline{L_n})$ be the pairs on which A and B conclusion-conclusion attack, where for $i = 1 \ldots n$, L_i and $\overline{L}_i$ are conclusions in A and B respectively. Then A is preferred$_\prec$ to B if for $i = 1 \ldots n$, $R_{L_i}(A + S_i) > R_{\overline{L}_i}(B + S'_i)$*

In example 2, $B3$ concludes $r2 \prec r1, r4 \prec r3$, and so $R_a(A3) > R_{\neg a}(A4)$, $R_b(A3) > R_{\neg b}(A4)$, and $A3$ is preferred$_\prec$ to $A4$. In ALP-DP, conclusion-conclusion attacks are preference dependent and (by definition) symmetric, and undercuts are preference independent. We can now instantiate a symmetric *EAF* with the arguments, their attacks, and priority arguments claiming preferences and so *d* attacking *pd* attacks:

Definition 16. *The* sEAF *$(Args, \mathcal{R}_{pd}, \mathcal{R}_{pi}, \mathcal{R}_d)$ for a theory (S, D) is defined as follows. $Args$ is the set of arguments given by definition 10, and $\forall A, B, C \in Args$:*

1. *(C,(B, A)) $\in \mathcal{R}_d$ iff C concludes $\prec$ and A is preferred$_\prec$ to B*
2. *(A, B),$(B, A) \in \mathcal{R}_{pd}$ iff A and B conclusion-conclusion attack*
3. *$(A, B) \in \mathcal{R}_{pi}$ iff A undercuts B*

Note that it automatically follows that if$(C,(B, A)), (C',(A, B)) \in \mathcal{R}_d$ then$(C, C'),(C', C) \in \mathcal{R}_{pd}$, since:

Proposition 6. *If C and C' respectively conclude $\prec$ and $\prec'$, and A is preferred$_\prec$ to B, B is preferred$_{\prec'}$ to A, then C and C' conclusion-conclusion attack.*

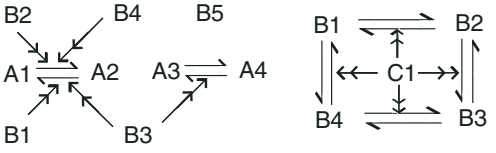


Fig. 3. The *sEAF* for example 2

In ALP-DP, an argument A cannot defeat B if B undercuts A. Hence, we assume the S-2-defeat relation (in definition 2) when evaluating the justified arguments of a theory's s*EAF* under all of the extensional semantics. For example 2's *sEAF* in fig.3, $\{C1, B1, B3, B5, A1, A3\}$ is a subset of the single grounded, preferred, complete and stable extension E. E additionally includes all composite arguments that can be constructed from $C1, B1, B3$, and $B5$ (so excluding self *pd* attacking arguments such as $B6$ = [r5, r6], whose sub-arguments are $B1$ and $B2$). This follows from the following:

Proposition 7. *A is sceptically/credulously justified iff all sub-arguments of A are sceptically/credulously justified.*

In [11], only the strict acceptability of A w.r.t. a set T is defined: every B that defeats A must be strictly defeated by a $C \in T$, where defeat is defined w.r.t. the ordering concluded by singleton priority arguments in T. This precludes ALP-DP from defining acceptability of sets of arguments under the admissible, complete and preferred semantics. In [11], the grounded extension of (S, D) is obtained by constructing the least fixed point of the theory's characteristic function defined in terms of strict acceptability. Indeed, given proposition 5, the following can be shown to hold:

Proposition 8. *Let G be the grounded extension of a theory (S, D) as defined in [11]. Let G' be the grounded extension of the theory's sEAF. Then:*

1. *$G \subseteq G'$*
2. *If $A \in G'$ and $A \notin G$ then A is a composite priority argument constructed from the singleton priority arguments $A_1, \ldots, A_n$, where $n > 1$ and for $i = 1 \ldots n$, $A_i \in G$*

Example 3. Consider a theory's arguments:
A = [$guardian :\Rightarrow sky \prec bbc$], B = [$sun :\Rightarrow bbc \prec sky$], C = [$bbc :\Rightarrow sun \prec guardian$], D = [$sky :\Rightarrow guardian \prec sun$], where $x :\Rightarrow y \prec z$ is interpreted as x states that y is less trustworthy than z. We have that:
$A \rightleftharpoons B, C \rightleftharpoons D, A \twoheadrightarrow (D \rightharpoonup C), B \twoheadrightarrow (C \rightharpoonup D), C \twoheadrightarrow (B \rightharpoonup A), D \twoheadrightarrow (A \rightharpoonup B)$
$\emptyset$ is the grounded extension, and $\{A, C\}$, $\{B, D\}$ the preferred extensions, each of which represent the mutually supportive media outlets. Recall that in section 1 we mentioned hierarchical argumentation frameworks [9] that would have difficulty formalising this kind of example because level n framework arguments cannot themselves contribute to level $n + 1$ reasoning about their relative preferences.

5 Conclusions and Future Work

In this paper we have extended Dung's abstract theory of argumentation to accommodate differing notions of conflict, and defeasible reasoning about preferences between

arguments. We suggested that existing non-monotonic formalisms facilitating defeasible reasoning about priorities in the object language, can be formalised and extended in our framework. To illustrate, we formalised and extended (by formalising acceptability under the full range of semantics) logic programming with defeasible priorities [11]. We also showed how works such as [4] that assume pre-defined preference (value) orderings on arguments, can now be extended to enable defeasible reasoning about preferences. We believe that other related works (e.g. [1]) can also be formalised and similarly extended in our framework. We also claim our work will facilitate development of new formalisms addressing requirements for application of argumentation to single agent reasoning, and argumentation based dialogues in which arguments supporting statements are exchanged and evaluated. Argumentation is being applied to resolution of variously defined notions of conflict arising within and amongst mental attitudes, including beliefs, desires, goals, intentions e.t.c. Works such as [8],[10] and [2] illustrate the need to reason about conflicting preferences arising in different contexts, as a result of multiple perspectives, and when different criteria are applied to evaluate argument strength. Finally, we mention current work on argument game proof theories for *EAFs*, whereby the credulous-preferred and sceptical-grounded acceptability of an argument A is determined by a regulated dispute between the proponent and opponent of A.

References

1. Amgoud, L.: Using Preferences to Select Acceptable Arguments. In: Proc. 13th European Conference on Artificial Intelligence, pp. 43–44 (1998)
2. Amgoud, L., Parsons, S.: Agent Dialogues with Conflicting Preferences. In: Meyer, J.-J.C., Tambe, M. (eds.) ATAL 2001. LNCS (LNAI), vol. 2333, pp. 190–205. Springer, Heidelberg (2002)
3. Atkinson, K.M., Bench-Capon, T.J.M., McBurney, P.: Computational Representation of Practical Argument. Synthese 152(2), 157–206 (2006)
4. Bench-Capon, T.J.M.: Persuasion in Practical Argument Using Value-based Argumentation Frameworks. Journal of Logic and Computation 13(3), 429–448 (2003)
5. Bondarenko, A., Dung, P.M., Kowalski, R.A., Toni, F.: An abstract, argumentation-theoretic approach to default reasoning. Artificial Intelligence 93, 63–101 (1997)
6. Brewka, G.: Reasoning about priorities in default logic. In: Proc. 12th National Conference on Artificial Intelligence (AAAI'94), pp. 940–945 (1994)
7. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77, 321–357 (1995)
8. Kakas, A., Moraitis, P.: Argumentation based decision making for autonomous agents. In: Proc. Second international joint conference on autonomous agents and multiagent systems, pp. 883–890 (2003)
9. Modgil, S.: Hierarchical Argumentation. In: Proc. 10th European Conference on Logics in Artificial Intelligence, pp. 319–332 (2006)
10. Modgil, S.: Value Based Argumentation in Hierarchical Argumentation Frameworks. In: Proc. 1st International Conference on Computational Models of Argument, pp. 297–308 (2006)
11. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. Journal of Applied Non-Classical Logics 7, 25–75 (1997)

Relaxing Ceteris Paribus Preferences with Partially Ordered Priorities

Souhila Kaci[1] and Henri Prade[2]

[1] C.R.I.L., Rue de l'Université SP 16 62307 Lens Cedex, France
[2] I.R.I.T., 118 route de Narbonne 31062 Toulouse Cedex 9, France

Abstract. Conditional preference networks (CP-nets) are a simple approach to the compact representation of preferences. In spite of their merit the application of the ceteris paribus principle underlying them is too global and systematic and sometimes leads to questionable incomparabilities. Moreover there is a natural need for expressing default preferences that generally hold, together with more specific ones that reverse them. This suggests the introduction of priorities for handling preferences in a more local way. After providing the necessary background on CP-nets and identifying the representation issues, the paper presents a logical encoding of preferences under the form of a partially ordered base of logical formulas using a discrimin ordering of the preferences. It is shown that it provides a better approximation of CP-nets than other approaches. This approximation is faithful w.r.t. the strict preferences part of the CP-net and enables a better control of the incomparabilites. Its computational cost remains polynomial w.r.t. the size of the CP-net. The case of cyclic CP-nets is also discussed.

Keywords: Conditional preferences, logical representation, reasoning about preferences, partial order, priority.

1 Introduction

Partially ordered information is naturally encountered in a variety of situations. Pieces of knowledge or belief may not be equally reliable, and their levels of reliability, which may be ordered in general when all the information comes from the same source, may become incomparable in case of several sources. When representing preferences, similar situations exist, especially in case of agents having different points of view. For representing such information, partial order relations have to be defined between outcomes, between formulas, and between sets of formulas in a compatible way (e. g., [11,1], etc). Various relations that may be more or less strong can be defined, which lead to different inference relations of various strength.

The expression of preferences is in practice a matter of context and priority. Conditional preferences networks (CP-nets for short; [2]) is a well-known framework for the compact representation of preferences, where contextual preferences are specified between incompletely specified situations in a *ceteris paribus* way, i.e. everything else being equal. From a CP-net, a strict partial order can be obtained between completely specified situations. Moreover, the preferential comparison of two completely specified situations is limited to the pairs for which it exists a path between them through

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 660–671, 2007.

a sequence of such situations, such that two successive situations differ only by one "worsening flip", the worsening being specified by the conditional preference tables of the CP-net under the ceteris paribus assumption. However, approximating a CP-net may be of interest in the general case for computational reasons. Besides, in spite of rather versatile expressive capabilities, CP-nets use a uniform expression format, which does not allow for the direct specification of default preferences. In [10], a possibilistic logic framework has been advocated where formulas are associated with priorities, for representing preferences. This setting uses complete preorders, which thus does not allow for incomparabilities, and its computational cost remains apparently high. However, the representation format allows for the specification of default preferences.

In the following we propose an original approach, based on partially ordered logical formulas, that allows for incomparability and default preferences, but which is still able to provide a faithful approximation of CP-nets, and which is computationally tractable.

After some definitions and notations given in Section 2, Section 3 first gives the necessary background on CP-nets, before discussing their expressive power. Then, the proposed approach is presented where the set of outcomes of a partially ordered set of logical formulas are rank-ordered on the basis of Brewka's preferred sub-theories [5]. In Section 4, an approximation of acyclic CP-nets in the logical framework is provided. The approximation power of the approach when giving priority to parent nodes enables to retrieve the associated CP-net-based partial order exactly, up to some incomparabilities that are turned into strict preferences. Section 5 points out the high quality of the approximation. Then the approach is shown to apply to cyclic CP-nets as well in Section 6, while Section 7 shows how the approach handles default preferences.

2 Definitions and Notations

Let $V = \{X_1, \cdots, X_l\}$ be a set of l variables. Each variable X_i takes its values in a domain denoted $Dom(X_i) = \{x_1^i, \cdots, x_{m_i}^i\}$. Let V' be a subset of V. An assignment of V' is the result of giving a value in $Dom(X_i)$ to each variable X_i in V'. $Asst(V')$ is the set of all possible assignments to variables in V'. In particular $Asst(V)$, denoted Ω, is the set of all possible assignments of the variables in V. Each element in Ω, denoted ω, is called an outcome. When dealing with binary variables, formulas of propositional logic are denoted $\varphi, \phi, \psi, \cdots$. $Mod(\varphi)$ denotes the set of outcomes satisfying φ.

Let $\succeq$ (resp. $\succ$) be a binary relation on a finite set $A = \{x, y, z, \cdots\}$ such that $x \succeq y$ (resp. $x \succ y$) means that x is at least as preferred as (resp. strictly preferred to) y. $x = y$ means that both $x \succeq y$ and $y \succeq x$ hold, i.e. x and y are equally preferred. Lastly $x \sim y$ means that neither $x \succeq y$ nor $y \succeq x$ holds, i.e. x and y are incomparable.

$\succeq$ is a partial preorder on A if and only if $\succeq$ is reflexive ($x \succeq x$) and transitive (if $x \succeq y$ and $y \succeq z$ then $x \succeq z$). $\succ$ is a partial order on A if and only if $\succ$ is irreflexive ($x \succ x$ does not hold) and transitive. A partial order $\succ$ may be defined from a partial preorder $\succeq$ as $x \succ y$ if $x \succeq y$ holds but $y \succeq x$ does not. A (pre-)order is asymmetric if and only if $\forall x, y \in A$, if $x \succ y$ holds then $y \succ x$ does not. A preorder $\succeq$ on A is complete if and only if all pairs are comparable i.e. $\forall x, y \in A$, we have $x \succeq y$ or $y \succeq x$.

The set of the best/undominated (resp. worst) elements of A w.r.t. $\succ$, denoted $\max(A, \succ)$ (resp. $\min(A, \succ)$), is defined by $\{x | x \in A, \nexists y \in A, y \succ x\}$ (resp. $\{x | x \in A, \nexists y \in$

$A, x \succ y\}$). The set of the best (resp. worst) elements of A w.r.t. a preorder $\succeq$ is $\max(A, \succ)$ (resp. $\min(A, \succ)$) where $\succ$ is the strict order associated to $\succeq$.

3 Partially Ordered Preference Statements

Different compact representations of partial preorders have been proposed in literature [2,5,14,1]. In this paper due to space limitation, we only focus on two representations.

3.1 Conditional Preference Networks

Conditional preference networks (CP-nets for short) [2] are based on comparative conditional statements, together with ceteris paribus principle. A CP-net is a directed graphical representation of conditional preferences, where nodes represent variables and edges express preference links between variables. When there exists a link from X to Y, X is called a parent of Y. $Pa(X)$ denotes the set of parents of a given node X. It determines the user's preferences over possible values of X. For the sake of simplicity, we suppose that variables are binary. Preferences are expressed at each node by means of a *conditional preference table* (CPT for short) such that:

- for root nodes X_i, the conditional preference table, denoted $CPT(X_i)$, provides the strict preference[1] over x^i and its negation $\neg x^i$, other things being equal, i.e. $\forall y \in Asst(Y), x^i y \succ \neg x^i y$ where $Y = V \backslash \{X_i\}$. This is the ceteris paribus principle.
- For other nodes X_j, $CPT(X_j)$ describes the preferences over x^j and $\neg x^j$ other things being equal given any assignment of $Pa(X_j)$, i.e. $x^j zy \succ \neg x^j zy$, $\forall z \in Asst(Pa(X_j))$ and $\forall y \in Asst(Y)$ where $Y = V \backslash (\{X_j\} \cup Pa(X_j))$. For each assignment z of $Pa(X_j)$ we write for short a statement of the form $z : x^j \succ \neg x^j$. Note that this is a parent-dependent specification.

We define the size of a CP-net $\mathcal{N}$ as the number of conditional/unconditional preferences expressed in $\mathcal{N}$.

Definition 1. *A complete preorder $\succeq$ on Ω, called also preference ranking,* satisfies *a CP-net $\mathcal{N}$ if and only if it satisfies each conditional preference expressed in $\mathcal{N}$. In this case, we say that the preference ranking $\succeq$ is consistent with $\mathcal{N}$.*

A CP-net $\mathcal{N}$ is consistent when there exists an asymmetric preference ranking that is consistent with $\mathcal{N}$. We mainly focus in this paper on acyclic CP-nets in order to ensure their consistency. The case of cyclic CP-nets is discussed in Section 6.

Definition 2 (Preference entailment). *Let $\mathcal{N}$ be a CP-net over a set of variables V, and $\omega, \omega' \in \Omega$. $\mathcal{N}$ entails that ω is strictly preferred to ω', denoted $\omega \succ_{\mathcal{N}} \omega'$, if and only if $\omega \succ \omega'$ holds in every preference ranking $\succeq$ that satisfies $\mathcal{N}$.*

[1] We restrict ourselves to a complete order over x^i and $\neg x^i$ as it is the case with CP-nets in general. However this can be easily extended to a preorder as it is the case in Section 5.

Indeed $\succ_{\mathcal{N}}$ is the intersection of all preference rankings consistent with $\mathcal{N}$. When $\omega \succ_{\mathcal{N}} \omega'$ holds, we say that ω dominates ω'. The set of optimal outcomes of a CP-net $\mathcal{N}$ is $\max(\Omega, \succ_{\mathcal{N}})$. The preferential comparison in CP-nets is based on the notion of *worsening* flip. A worsening flip is a change of the value of one variable in an assignment in such a way that the new assignment is less preferred according to the conditional preference table of that variable, and under ceteris paribus assumption, w.r.t. the CP-net $\mathcal{N}$. Then ω is preferred to ω' w.r.t. $\mathcal{N}$ iff there is a chain of worsening flips from ω to ω'. For instance $V_bP_bS_rC_r \succ_{\mathcal{N}} V_wP_bS_wC_w$ because we have $V_bP_bS_rC_r \succ_{\mathcal{N}} V_bP_bS_rC_w \succ_{\mathcal{N}} V_bP_bS_wC_w \succ_{\mathcal{N}} V_wP_bS_wC_w$, in the example below.

Example 1. (borrowed from [9]) Let $\mathcal{N}$ be a CP-net over a set of binary variables V (vest), P (pant), S (shirt) and C (shoes) s.t. $Dom(V) = \{V_b, V_w\}$, $Dom(P) = \{P_b, P_w\}$, $Dom(S) = \{S_r, S_w\}$ and $Dom(C) = \{C_r, C_w\}$, where b, w and r stand respectively for black, white and red. Fig. 1 gives the structure of $\mathcal{N}$ together with its associated partial order. For the sake of readability we did not represent in this latter

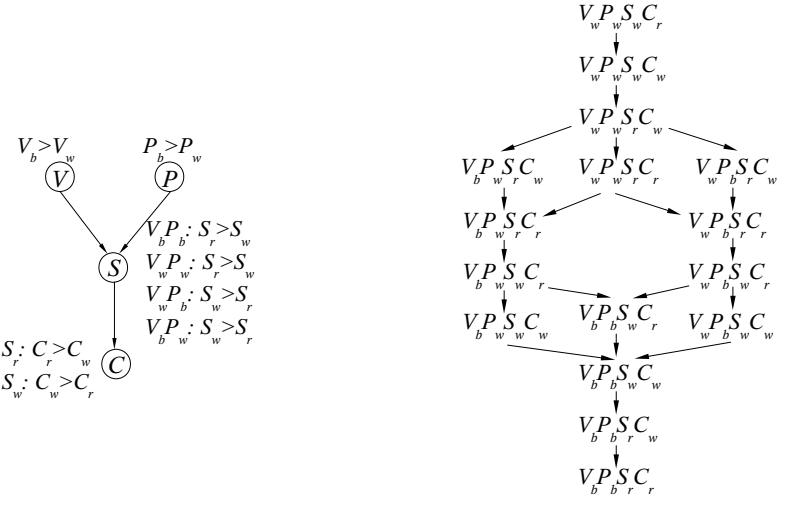


Fig. 1. Example of CP-net

strict preferences induced by $\mathcal{N}$ that can also be deduced by transitivity. An edge from an outcome ω to an outcome ω' means that ω' is preferred to ω, i.e. $\omega' \succ_{\mathcal{N}} \omega$.

TCP-nets (T for tradeoff) [4] are an extension of CP-nets with variable importance tradeoffs. They allow to express a (conditional) relative importance of a variable X over another variable Y, interpreted as: "it is more important for us to see X getting its most preferred value than to see Y getting its most preferred value". For example V is more important than P leads to the following additional preference statements: $V_bP_wS_rC_r \succ V_wP_bS_rC_r$, $V_bP_wS_rC_w \succ V_wP_bS_rC_w$, $V_bP_wS_wC_r \succ V_wP_bS_wC_r$ and $V_bP_wS_wC_w \succ V_wP_bS_wC_w$.

In CP-theories [14] (non graphical representations), preference statements are of the form $u : x \succ x'[S]$, where $S \subseteq V \backslash (\{X\} \cup Pa(X))$. Let $T = V \backslash (S \cup \{X\} \cup Pa(X))$. The preference statement $u : x \succ x'[S]$ stands for "in the context u, x is preferred to x' irrespective to the values of S, other things being equal", i.e. $\forall t \in Asst(T), \forall s, s' \in Asst(S), tuxs \succ tux's'$, where $u \in Asst(Pa(X))$. Wilson [14] has shown that CP-nets and TCP-nets are special cases of CP-theories.

In the above example, due to the structure of the CP-net and its associated preference tables we have that $V_b P_b S_r C_w \succ_{\mathcal{N}} V_w P_w S_r C_w$, $V_b P_w S_r C_w \succ_{\mathcal{N}} V_w P_w S_r C_w$ and $V_w P_b S_r C_w \succ_{\mathcal{N}} V_w P_w S_r C_w$. But one may like to prefer V_w and P_w when S_r is true but not C_r. More precisely we have $S_r \wedge C_w : V_w \wedge P_w \succ V_b \vee P_b$. This means that we prefer V_b to V_w, P_b to P_w unless $S_r \wedge C_w$ is true in which case our preferences over values of V and P are reversed. This situation cannot be captured by TCP-nets since they allow the expression of priority between conditionally independent variables only. The preference statement $S_r \wedge C_w : V_w \wedge P_w \succ V_b \vee P_b$ is more specific than both $\top : V_b \succ V_w$ and $\top : P_b \succ P_w$. Technically speaking, $S_r \wedge C_w : V_w \wedge P_w \succ V_b \vee P_b$ should be given a higher priority over both $\top : V_b \succ V_w$ and $\top : P_b \succ P_w$. In such a way we prefer $V_w \wedge P_w$ when $S_r \wedge C_w$ is true and V_b and P_b in all other cases. The principle here is to apply default reasoning to preference modeling, i.e. we have general preferences ($\top : V_b \succ V_w$ and $\top : P_b \succ P_w$) and a specific one $S_r \wedge C_w : V_w \wedge P_w \succ V_b \vee P_b$.

Many approaches to default reasoning take advantage of an ordering for properly handling specific contexts where plausible conclusions differ from the general ones. The above concerns thus suggest the introduction of priorities associated with logical formulas encoding preferences for introducing more flexibilities in their specification.

3.2 Partially Ordered Formulas

We represent prioritized formulas by means of a pair $(\Sigma, \succeq_\Sigma)$ where Σ is a set of propositional formulas and $\succeq_\Sigma$ is a preorder on Σ. When $\succeq_\Sigma$ is a partial preorder, different criteria have been proposed in literature to rank-order the set of outcomes [11,5,1]. For the purpose of this paper we only focus on Brewka's proposal [5].

Definition 3 (Discrimin preferences). *Let $(\Sigma, \succeq_\Sigma)$ be a partially ordered base. Let $\omega, \omega' \in \Omega$. Let $\mathcal{F}_\omega = \{\varphi \mid \varphi \in \Sigma, \omega \not\models \varphi\}$ and $\mathcal{F}_{\omega'} = \{\varphi \mid \varphi \in \Sigma, \omega' \not\models \varphi\}$.*

- *$\omega \succ_{\Omega,d} \omega'$ iff $\forall \varphi \in \mathcal{F}_\omega \backslash \mathcal{F}_{\omega'}$, $\exists \varphi' \in \mathcal{F}_{\omega'} \backslash \mathcal{F}_\omega$ such that $\varphi' \succ_\Sigma \varphi$.*
- *$\omega \succeq_{\Omega,d} \omega'$ iff $\forall \varphi \in \mathcal{F}_\omega \backslash \mathcal{F}_{\omega'}$, $\exists \varphi' \in \mathcal{F}_{\omega'} \backslash \mathcal{F}_\omega$ such that $\varphi' \succeq_\Sigma \varphi$.*
- *$\omega =_{\Omega,d} \omega'$ iff both $\omega \succeq_{\Omega,d} \omega'$ and $\omega' \succeq_{\Omega,d} \omega$ hold.*
  When $\mathcal{F}_\omega \subset \mathcal{F}_{\omega'}$ we have $\omega \succ_{\Omega,d} \omega'$.

Example 2. Let A, B, C and D be four binary variables. Let $(\Sigma, \succeq_\Sigma)$ with $\Sigma = \{a, b, c, d\}$ and $\succeq_\Sigma = \{b =_\Sigma c, b \succ_\Sigma d, c \succ_\Sigma d, b \succ_\Sigma a, c \succ_\Sigma a\}$. Let $\omega_0 = ab\neg cd$, $\omega_1 = a\neg bcd$, $\omega_2 = abc\neg d$, $\omega_3 = \neg abcd$, $\omega_4 = \neg ab\neg cd$, $\omega_5 = a\neg b\neg cd$. We have $\mathcal{F}_{\omega_0} = \{c\}$, $\mathcal{F}_{\omega_1} = \{b\}$, $\mathcal{F}_{\omega_2} = \{d\}$, $\mathcal{F}_{\omega_3} = \{a\}$, $\mathcal{F}_{\omega_4} = \{a, c\}$ and $\mathcal{F}_{\omega_5} = \{b, c\}$. Then $\omega_0 =_{\Omega,d} \omega_1$, $\omega_2 \sim_{\Omega,d} \omega_3$ and $\omega_4 \succ_{\Omega,d} \omega_5$.

While in knowledge representation the main problem is the inference, preference representation is concerned with comparing alternatives. The comparison of two alternatives following Definition 3 is achieved in polynomial time [12].

4 Approximation of Acyclic CP-Nets

Because of the exponential computational cost of dominance queries in CP-nets, several authors have made proposals for approximating CP-nets (as further discussed in Section 5). Thus far a good (or faithful) approximation of a CP-net should only recover all strict comparisons of the CP-net [9,6]. In this paper we strengthen the faithfulness criteria by leaving open the possibility of recovering incomparabilities. We approximate a CP-net by means of a set of partially ordered formulas that generates a partial preorder on the set of outcomes following Definition 3. This approximation follows three steps:

1. Let X be a node in the CP-net $\mathcal{N}$ and $CPT(X)$ be its associated conditional preference table. For each assignment of $Pa(X)$, $u_i : x \succ \neg x$ in $CPT(X)$ we associate a base made of one formula $\neg u_i \vee x$ as follows $\Sigma_{X,u_i} = \{\neg u_i \vee x\}$.
2. For each node X in the CP-net $\mathcal{N}$, build $\Sigma_X = \bigcup_i \Sigma_{X,u_i}$ where the bases Σ_{X,u_i} have been obtained at the previous step. Then $\Sigma = \bigcup_X \Sigma_X$ is the partially ordered base associated with $\mathcal{N}$.
3. A partial preorder $\succeq_\Sigma$ on Σ is defined where each formula associated to a node is strictly preferred to all formulas associated to its child nodes[2]. The idea underlying this step is simple, any formula associated with a node is more important than formulas associated with any child node. This is illustrated on our running example.

Example 3. (Ex. 1 cont'd) $\Sigma = \{V_b, P_b, \neg V_w \vee \neg P_w \vee S_r, \neg V_b \vee \neg P_b \vee S_r, \neg V_b \vee \neg P_w \vee S_w, \neg V_w \vee \neg P_b \vee S_w, \neg S_w \vee C_w, \neg S_r \vee C_r\}$ with $\succeq_\Sigma = \{V_b \succ_\Sigma \neg V_w \vee \neg P_w \vee S_r$, $V_b \succ_\Sigma \neg V_b \vee \neg P_b \vee S_r$, $V_b \succ_\Sigma \neg V_b \vee \neg P_w \vee S_w$, $V_b \succ_\Sigma \neg V_w \vee \neg P_b \vee S_w$, $P_b \succ_\Sigma \neg V_w \vee \neg P_w \vee S_r$, $P_b \succ_\Sigma \neg V_b \vee \neg P_b \vee S_r$, $P_b \succ_\Sigma \neg V_b \vee \neg P_w \vee S_w$, $P_b \succ_\Sigma \neg V_w \vee \neg P_b \vee S_w, \neg V_w \vee \neg P_w \vee S_r \succ_\Sigma \neg S_w \vee C_w, \neg V_w \vee \neg P_w \vee S_r \succ_\Sigma \neg S_r \vee C_r$, $\neg V_b \vee \neg P_b \vee S_r \succ_\Sigma \neg S_w \vee C_w, \neg V_b \vee \neg P_b \vee S_r \succ_\Sigma \neg S_r \vee C_r$, $\neg V_b \vee \neg P_w \vee S_w \succ_\Sigma \neg S_w \vee C_w, \neg V_b \vee \neg P_w \vee S_w \succ_\Sigma \neg S_r \vee C_r$, $\neg V_w \vee \neg P_b \vee S_w \succ_\Sigma \neg S_w \vee C_w, \neg V_w \vee \neg P_b \vee S_w \succ_\Sigma \neg S_r \vee C_r\}$.

See Fig. 2. An edge from $\psi \sim \psi'$ to $\varphi \sim \varphi'$ means that φ and φ' (resp. ψ and ψ') are incomparable w.r.t. $\succeq_\Sigma$ but each one is strictly preferred to both ψ and ψ' w.r.t. $\succeq_\Sigma$.

Remark 1. For the sake of simplicity, we focused in the above approximation on preference statements of the form $u : x \succ \neg x$ used with CP-nets in general. This can be extended to cover all other cases. If $u : x \sim \neg x$ then we add $\neg u \vee x$ and $\neg u \vee \neg x$ to Σ. If $u : x \succeq \neg x$ (resp. $u : x = \neg x$) then we add $\neg u \vee x$ and $\neg u \vee \neg x$ to Σ, and $\neg u \vee x \succeq_\Sigma \neg u \vee \neg x$ (resp. $\neg u \vee x =_\Sigma \neg u \vee \neg x$) to $\succeq_\Sigma$. However a CP-net based on $\succeq$ or $=$ statements may be inconsistent even if it is acyclic. An example is given in Section 5. For this reason we focus in this section on strict preference statements together with the initial hypothesis that CP-nets are acyclic.

The following proposition can be shown, which expresses that the partially ordered base constructed from a CP-net $\mathcal{N}$ satisfies *all ceteris paribus* preferences induced by $\mathcal{N}$.

[2] Since usually CP-nets use local strict preference relations, $\succeq_\Sigma$ is a partial order but we speak about a partial preorder in order to keep the definitions general. See Section 5.

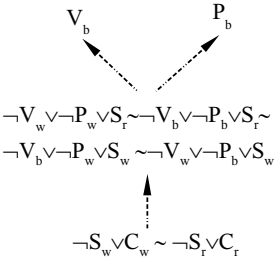


Fig. 2. Partial preorder associated to Σ

Proposition 1 (Ceteris paribus preferences). *Let $\mathcal{N}$ be an acyclic CP-net over a set of variables V. Let X be a node in $\mathcal{N}$ and $Y = V \backslash (\{X\} \cup Pa(X))$. Let $(\Sigma, \succeq_\Sigma)$ be the partially ordered base associated with $\mathcal{N}$ and $\succ_{\Omega,d}$ be the partial order associated to $(\Sigma, \succeq_\Sigma)$ following Definition 3. Let $u : x_1 \succ x_2$ be a preference stated by $\mathcal{N}$ where u is an assignment of $Pa(X)$. Then for each assignement y of Y we have $ux_1y \succ_{\Omega,d} ux_2y$.*

Example 4. (Example 1 continued) Let us consider the node S and the preference statement $V_w \wedge P_b : S_w \succ S_r$. Following ceteris paribus principle we have $V_wP_bS_wC_r \succ_\mathcal{N} V_wP_bS_rC_r$ and $V_wP_bS_wC_w \succ_\mathcal{N} V_wP_bS_rC_w$. Let us now compare these outcomes w.r.t. $(\Sigma, \succeq_\Sigma)$. Let $\omega_4 = V_wP_bS_wC_w$, $\omega_5 = V_wP_bS_wC_r$, $\omega_6 = V_wP_bS_rC_w$ and $\omega_7 = V_wP_bS_rC_r$. We have $\mathcal{F}_{\omega_5} = \{V_b, \neg S_w \vee C_w\}$ and $\mathcal{F}_{\omega_7} = \{V_b, \neg V_w \vee \neg P_b \vee S_w\}$. $\mathcal{F}_{\omega_5} \backslash \mathcal{F}_{\omega_7} = \{\neg S_w \vee C_w\}$ and $\mathcal{F}_{\omega_7} \backslash \mathcal{F}_{\omega_5} = \{\neg V_w \vee \neg P_b \vee S_w\}$. We have $\neg V_w \vee \neg P_b \vee S_w \succ_\Sigma \neg S_w \vee C_w$ so $\omega_5 \succ_{\Omega,d} \omega_7$. Now $\mathcal{F}_{\omega_4} = \{V_b\}$ and $\mathcal{F}_{\omega_6} = \{V_b, \neg V_w \vee \neg P_b \vee S_w, \neg S_r \vee C_r\}$. We have $\mathcal{F}_{\omega_4} \subset \mathcal{F}_{\omega_6}$ so $\omega_4 \succ_{\Omega,d} \omega_6$.

It follows from Proposition 1 that the partially ordered base associated with $\mathcal{N}$ recovers *all strict preferences* induced by $\mathcal{N}$. Formally we have:

Proposition 2 (Strict preferences). *Let $\mathcal{N}$ be an acyclic CP-net and $(\Sigma, \succeq_\Sigma)$ be its associated partially ordered base. Let $\succ_{\Omega,d}$ be the partial order associated to $(\Sigma, \succeq_\Sigma)$ following Definition 3. Then, $\forall \omega, \omega' \in \Omega$, if $\omega \succ_\mathcal{N} \omega'$ then $\omega \succ_{\Omega,d} \omega'$.*

Example 5. (Example 1 continued) Let $\omega_2 = V_wP_wS_rC_w$ and $\omega_7 = V_wP_bS_rC_r$. Following $\mathcal{N}$ we have $\omega_7 \succ_\mathcal{N} \omega_2$. We also have $\mathcal{F}_{\omega_2} = \{V_b, P_b, \neg S_r \vee C_r\}$ and $\mathcal{F}_{\omega_7} = \{V_b, \neg V_w \vee \neg P_b \vee S_w\}$. $\mathcal{F}_{\omega_2} \backslash \mathcal{F}_{\omega_7} = \{P_b, \neg S_r \vee C_r\}$ and $\mathcal{F}_{\omega_7} \backslash \mathcal{F}_{\omega_2} = \{\neg V_w \vee \neg C_b \vee S_w\}$. We have $P_b \succ_\Sigma \neg V_w \vee \neg P_b \vee S_w$ so $\omega_7 \succ_{\Omega,d} \omega_2$. Fig. 3 gives the partial order associated with $(\Sigma, \succeq_\Sigma)$ following Definition 3.

Let us now examine what kind of comparisons of outcomes the partially ordered base constructed above offers – except the fact already established in Proposition 2 that it preserves all strict preferences induced by the CP-net.

Proposition 3. *Let $\mathcal{N}$ be an acyclic CP-net and $(\Sigma, \succeq_\Sigma)$ be its associated partially ordered base. Let $\succ_{\Omega,d}$ be the partial order associated with $(\Sigma, \succeq_\Sigma)$ following Def. 3.*

- *$\forall \omega, \omega' \in \Omega$, if $\omega \succ_{\Omega,d} \omega'$ then ($\omega \succ_\mathcal{N} \omega'$ or $\omega \sim_\mathcal{N} \omega'$),*
- *$\forall \omega, \omega' \in \Omega$, if $\omega \sim_{\Omega,d} \omega'$ then $\omega \sim_\mathcal{N} \omega'$.*

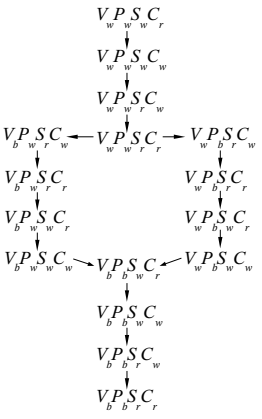


Fig. 3. Partial order associated to $(\Sigma, \succeq_\Sigma)$

It follows from Proposition 3 that our approximation recovers some incomparabilities induced by the CP-net. This result shows the good behavior of our approximation since it not only recovers all strict comparabilities of the CP-net but also some incomparabilities (existing approximations proposed in literature only focus on recovering strict comparabilities as further discussed in Section 5). But this result also raises natural questions: why can we not recover *all* incomparabilities of the CP-net since we are using a partial preorder? Is it a limitation of the proposed approximation? We think that this is not necessarily a limitation of our approximation. Rather it raises the question of the representation of ceteris paribus preferences in CP-nets, where incomparability remains between outcomes if one cannot go from one to the others by a succession of one-variable flips for which strict preference holds w.r.t. the CP-net. For instance, in our example of Fig. 1, the two outcomes $V_b P_b S_w C_r$ and $V_w P_b S_w C_w$ are incomparable. However, one may wonder if $V_b P_b S_w C_r$ should not be given priority over $V_w P_b S_w C_w$ since $V_b P_b S_w C_r$ falsifies preferences associated to child nodes S and C while $V_w P_b S_w C_w$ falsifies a preference associated to a parent node V. Indeed what happens on parent nodes determines the preference at the child node level. Lastly, note that our approximation doesn't need any extra effort in time and space.

Proposition 4. *Let $\mathcal{N}$ be an acyclic CP-net. The complexity of constructing $(\Sigma, \succeq_\Sigma)$ and comparing ω and ω' w.r.t. $(\Sigma, \succeq_\Sigma)$ is polynomial in the size of $\mathcal{N}$.*

The idea of giving priority to parent nodes over child nodes has been advocated in [2] and used in order to compute a complete order consistent with the CP-net. Our approximation is also based on this idea. However we use a formal logical setting for preference modeling. This allows to reason on general and specific preferences and is compatible with ceteris paribus principle (whose application is no longer compulsory anyway), as shown later in this paper.

5 Related Works

Our approximation provides a partial preorder $\succeq_{\Omega,d}$ that recovers all strict preferences induced by a CP-net offering a faithful approximation of CP-nets. It also recovers some incomparabilities of the CP-net. Moreover all incomparabilities w.r.t. $\succeq_{\Omega,d}$ are true in the CP-net, which means that our approximation is precise when two outcomes are incomparable. Other approximations of CP-nets have also been proposed based on weighted CSP [9] or answer sets optimization [6]. In these approaches a complete preorder is generated that recovers only strict preferences generated by the CP-net. In case of CP-nets associated with strict partial orders, equalities of the *complete* preorder in those approaches turn out to incomparablities in our approximation and conversely (see also [10]). However since our approximation provides a *partial* preorder, it is closer to CP-nets since it is more faithful to the incomparabilities in CP-nets. Let us consider the CP-net depicted in Fig. 4.a. The partial preorder associated to this CP-net is given in Fig. 4.b. Following the approximation given in [9] based on weighted CSP[3], we associate a weight to preferences associated to each node: the weight 1 (resp. 2 and 4) to preferences of the node C (resp. S, (V and P)). Then we compute the penalty of each outcome which is equal to the sum of weights of preferences that it falsifies. For example penalty of $V_wP_bS_rC_w$ is 7 because $V_wP_bS_rC_w$ falsifies $V_b \succ V_w$, $V_wP_b : S_w \succ S_r$ and $S_r : C_r \succ C_w$. Indeed we get the complete preorder given in Fig. 4.c. Using our approximation, we add $\neg S_w \vee C_r$ to Σ and $\neg S_w \vee C_w =_\Sigma \neg S_w \vee C_r$ to $\succeq_\Sigma$, where $(\Sigma, \succeq_\Sigma)$ is the partially ordered base given in Example 3. The associated partial preorder is given in Fig. 4.d. We can see that our approximation is more precise since it distinguishes incomparability and equality in the CP-net in contrast to [9] where incomparability and equality are confused. For instance $V_bP_wS_wC_r$, $V_bP_wS_wC_w$, $V_wP_bS_wC_r$ and $V_wP_bS_wC_w$ are equally preferred w.r.t. the weighted CSP, while we only have $V_bP_wS_wC_r = V_bP_wS_wC_w$ and $V_wP_bS_wC_r = V_wP_bS_wC_w$ w.r.t. both the CP-net and our approximation.

It is worth noticing that indifference in CP-nets may lead to an inconsistent CP-net even if the graph is acyclic. Let $\mathcal{N}'$ be a CP-net over two variables A and B such that A is a parent of B. Suppose that $a = \neg a$, $a : b \succ \neg b$ and $\neg a : \neg b \succ b$ [2]. Then we have $ab \succ_{\mathcal{N}'} a\neg b =_{\mathcal{N}'} \neg a\neg b \succ_{\mathcal{N}'} \neg ab =_{\mathcal{N}'} ab$. We could use our approximation on the example given in Fig. 4.a because the CP-net is consistent. Proposition 2 still holds when a CP-net is consistent and based on preference statements of the form $\succ$ and $=$. Moreover our approximation recovers all equalities of the CP-net.

A CP-net preference statement of the form $u : x \succ \neg x$ is encoded in our approximation by means of a formula $\neg u \vee x$. At the semantic level, this is interpreted as each outcome satisfying ux is preferred to each outcome satisfying $u\neg x$, i.e. $\forall y, y' \in Asst(Y), \forall t, t' \in Asst(T)$ with $Y, T \subset V \backslash (\{X\} \cup Pa(X))$, $Y \cap T = \emptyset$ and $Y \cup T \cup \{X\} \cup Pa(X) = V$, we have $uxyt \succ_{\Omega,d} u\neg xy't'$. If $y = y'$ and $t = t'$ then ceteris paribus preferences are recovered. If y and y' are any assignments in $Asst(Y)$ and $t = t'$ then we recover preference statements of CP-theories [14]. Indeed our ap-

[3] Due to the lack of space we only recall the approximation proposed in [9]. Note however that it is equivalent to the one proposed in [6]. Indeed the comparison presented in this section holds also for that work.

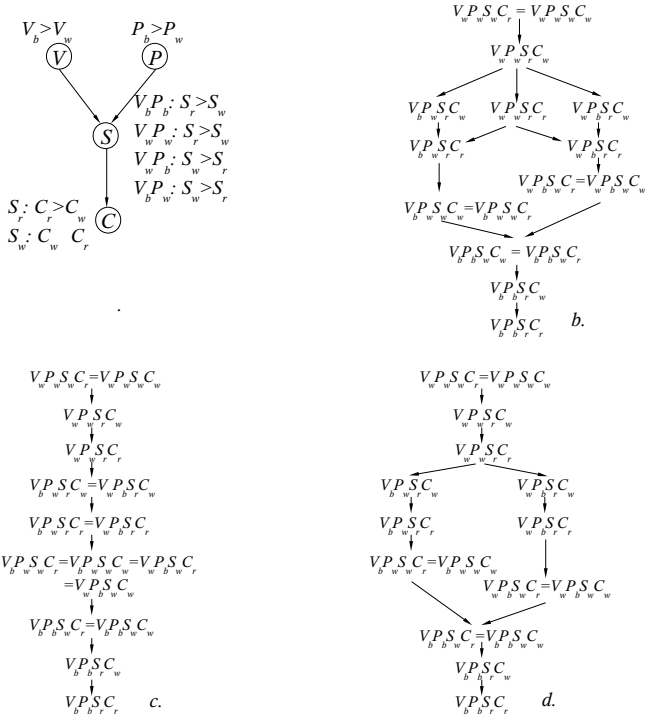


Fig. 4. A CP-net associated with a partial *pre*order

proximation also holds for CP-theories. The comparison of our approximation applied to CP-theories and the one proposed in [15] is left for a further research.

Lastly McGeachie and Doyle [13] use numerical function values to model ceteris paribus preferences, however their approach is computationally costly in general.

6 Cyclic CP-Nets

Brafman and Dimopoulos [3] addressed the case of cyclic CP-nets. Usually when a CP-net is cyclic and inconsistent, we conclude that there is no optimal outcomes. However in some cases, optimal solutions exist despite the fact that the CP-net is inconsistent. The CP-net depicted in Fig. 5 is an example of such a case. The CP-net is inconsistent but the outcomes $a\neg bc\neg d$ and $\neg ab\neg cd$ are not dominated. Also $abcd$ and $\neg a\neg b\neg c\neg d$ are the worst outcomes. In order to overcome this problem, the authors replace the strict preference relation $\succ$ in $CPT(.)$ by $\succeq$. Indeed five classes of outcomes are obtained and the following preorder is derived: $a\neg bc\neg d \sim_{\mathcal{N}''} \neg ab\neg cd \succ_{\mathcal{N}''} a\neg bcd =_{\mathcal{N}''} a\neg b\neg cd =_{\mathcal{N}''} a\neg b\neg c\neg d =_{\mathcal{N}''} ab\neg c\neg d =_{\mathcal{N}''} ab\neg cd =_{\mathcal{N}''} abc\neg d =_{\mathcal{N}''} \neg a\neg bcd =_{\mathcal{N}''} \neg a\neg b\neg cd =_{\mathcal{N}''} \neg a\neg bc\neg d =_{\mathcal{N}''} \neg ab\neg c\neg d =_{\mathcal{N}''} \neg abcd =_{\mathcal{N}''} \neg abc\neg d \succ_{\mathcal{N}''} abcd \sim_{\mathcal{N}''} \neg a\neg b\neg c\neg d$. This relaxation allows to recover best and worst outcomes.

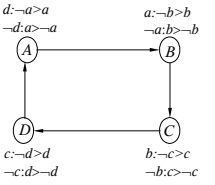


Fig. 5. A cyclic CP-net $\mathcal{N}''$

In contrast to this approach, our approximation does not need any relaxation of $\succ$. We have $\Sigma'' = \{\neg d \vee \neg a, d \vee a, \neg a \vee \neg b, a \vee b, \neg b \vee \neg c, b \vee c, \neg c \vee \neg d, c \vee d\}$ and $\succeq_{\Sigma''} = \{\neg d \vee \neg a \sim_{\Sigma''} d \vee a \succ_{\Sigma''} \neg a \vee \neg b \sim_{\Sigma''} a \vee b, \neg a \vee \neg b \sim_{\Sigma''} a \vee b \succ_{\Sigma''} \neg b \vee \neg c \sim_{\Sigma''} b \vee c, \neg b \vee \neg c \sim_{\Sigma''} b \vee c \succ_{\Sigma''} \neg c \vee \neg d \sim_{\Sigma''} c \vee d, \neg c \vee \neg d \sim_{\Sigma''} c \vee d \succ_{\Sigma''} \neg d \vee \neg a \sim_{\Sigma''} d \vee a\}$. We use our approximation previously described. The only difference is that the preference relation on formulas of Σ'' is no longer transitive but $\succeq_{\Omega,d}$ is still transitive. For example the outcomes $\omega_7 = \neg abcd$ and $\omega_{11} = a\neg bcd$ are not directly comparable because we have $\mathcal{F}_{\omega_7} \backslash \mathcal{F}_{\omega_{11}} = \{\neg b \vee \neg c\}$ and $\mathcal{F}_{\omega_{11}} \backslash \mathcal{F}_{\omega_7} = \{\neg d \vee \neg a\}$ and neither $\neg d \vee \neg a \succ_{\Sigma''} \neg b \vee \neg c$ nor $\neg b \vee \neg c \succ_{\Sigma''} \neg d \vee \neg a$ belongs explicitly to $\succeq_{\Sigma''}$. Let $\omega_8 = a\neg b\neg c\neg d$ and $\omega_9 = a\neg b\neg cd$. We have that ω_9 is preferred to ω_8 because $\mathcal{F}_{\omega_8} \backslash \mathcal{F}_{\omega_9} = \{c \vee d\}$, $\mathcal{F}_{\omega_9} \backslash \mathcal{F}_{\omega_8} = \{\neg d \vee \neg a\}$ and $c \vee d \succ_{\Sigma''} \neg d \vee \neg a$ belongs to $\succeq_{\Sigma''}$. Following Definition 3 the best outcomes w.r.t. $(\Sigma'', \succeq_{\Sigma''})$ are $\neg ab\neg cd$ and $a\neg bc\neg d$ while the worst ones are $\neg a\neg b\neg c\neg d$ and $abcd$.

7 General and Specific Preferences: An Example

Let us go back to the example presented in Section 3.1 that cannot be modelled by a CP-net. This example can be handled in our setting by giving priority to the specific preference $S_r \wedge C_w : V_w \wedge P_w \succ V_b \vee P_b$ over general preferences $\top : V_b \succ V_w$ and $\top : P_b \succ P_w$. Using our approximation, we can state this priority and provide a partial preorder that approximates the original CP-net extended with this additional priority. More precisely we have $(\Sigma', \succeq_{\Sigma'})$ where $\Sigma' = \Sigma \cup \{\neg S_r \vee \neg C_w \vee (V_w \wedge P_w)\}$ and $\succeq_{\Sigma'} = \succeq_{\Sigma} \cup \{\neg S_r \vee \neg C_w \vee (V_w \wedge P_w) \succ_{\Sigma} V_b, \neg S_r \vee \neg C_w \vee (V_w \wedge P_w) \succ_{\Sigma} P_b\}$, where $(\Sigma, \succeq_{\Sigma})$ is given in Example 3. Following Definition 3, the partial preorder associated to $(\Sigma', \succeq_{\Sigma'})$ is $V_bP_bS_rC_r \succ_{\Omega,d} V_bP_bS_wC_w \succ_{\Omega,d} V_bP_bS_wC_r \succ_{\Omega,d} V_bP_wS_wC_w \sim_{\Omega,d} V_wP_bS_wC_w \succ_{\Omega,d} V_wP_bS_wC_r \sim_{\Omega,d} V_bP_wS_wC_r \succ_{\Omega,d} V_bP_wS_rC_r \sim_{\Omega,d} V_wP_bS_rC_r \succ_{\Omega,d} V_wP_wS_rC_r \succ_{\Omega,d} V_wP_wS_rC_w \succ_{\Omega,d} V_wP_wS_wC_w \succ_{\Omega,d} V_wP_wS_wC_r \succ_{\Omega,d} V_bP_bS_rC_w \succ_{\Omega,d} V_bP_wS_rC_w \sim_{\Omega,d} V_wP_bS_rC_w$. We can check that in the context $S_r \wedge C_w$, $V_w \wedge P_w$ is preferred otherwise the ceteris paribus principle holds except when an outcome violates a preference associated with a parent node.

Dimopoulos et al. [8] proposed an extension of TCP-nets where a variable can be more important than its ancestors. This priority can be recovered in our framework by stating that preferences associated with the child node are preferred to those associated with its ancestors. Our framework is more general since the priority of a child over its ancestors can be restricted to some local preferences only.

8 Conclusion

The present work points out some expressiveness limitations of CP-nets and proposes a logical framework for preference representation. The proposed logical representation of partially ordered information offers a flexible manner for specifying preferences. Indeed, in this approach there is a complete freedom on the priority that can be put on formulas, and no such implicit priority in favor of parent nodes as in CP-nets.

Our proposed approach emphasizes issues related to the expressiveness of the representation, still remaining computationally tractable. The expressiveness issues raised in this paper offer a basis of comparison criteria to be considered together with computational complexity as in [7] when discussing the relative merits of different formalisms.

Acknowledgments

The first author has partially been supported by the French National Research Agency (Agence Nationale de la Recherche) under contract n^0 NT05-4_41833.

References

1. Benferhat, S., Lagrue, S., Papini, O.: Reasoning with partially ordered information in a possibilistic logic Framework. Fuzzy Sets and Systems 144(1), 25–41 (2004)
2. Boutilier, C., Brafman, R., Domshlak, C., Hoos, H., Poole, D.: CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. JAIR 21, 135–191 (2004)
3. Brafman, R., Dimopoulos, Y.: A new look at the semantics and optimization methods of CP-networks. Computational Intelligence 20(2), 137–157 (2004)
4. Brafman, R.I., Domshlak, C.: Introducing variables importance tradeoffs into CP-Nets. In: UAI'02, pp. 69–76 (2002)
5. Brewka, G.: Preferred subtheories: an extended logical framework for default reasoning. In: IJCAI'89, pp. 1043–1048 (1989)
6. Brewka, G., Niemelä, I., Truszczynski, M.: Answer set optimization. In: IJCAI'03, pp. 867–872 (2003)
7. Coste-Marquis, S., Lang, J., Liberatore, P., Marquis, P.: Expressive power and succinctness of propositional languages for preference representation. In: KR'04, pp. 203–212 (2004)
8. Dimopoulos, Y., Moraitis, P., Tsoukias, A.: Extending variable importance in preference networks. In: IJCAI'05 Workshop on Preferences (2005)
9. Domshlak, C., Venable, B., Rossi, F., Walsh, T.: Reasoning about soft constraints and conditional preferences. In: IJCAI'03, pp. 215–220 (2003)
10. Dubois, D., Kaci, S., Prade, H.: CP-nets and possibilistic logic: Two approaches to preference modeling. Steps towards a comparison. In: IJCAI'05 Workshop on Preferences (2005)
11. Halpern, J.: Defining relative likelihood in partially-ordered preferential structure. JAIR 7, 1–24 (1997)
12. Lang, J.: Logical preference representations and combinatorial vote. Annals on Mathematics and Artificial Intelligence 42(1-3), 37–71 (2004)
13. McGeachie, M., Doyle, J.: Efficient Utility Functions for Ceteris Paribus Preferences. In: AAAI'02, pp. 279–284 (2002)
14. Wilson, N.: Extending CP-nets with stronger conditional preference statements. In: AAAI'04, pp. 735–741 (2004)
15. Wilson, N.: An efficient upper approximation for conditional preference. In: ECAI'06, pp. 472–476 (2006)

Conceptual Uncertainty and Reasoning Tools

Bertil Rolf

Blekinge Institute of Technology, School of Management
SE-372 25 Ronneby, Sweden
bertil.rolf@bth.se

Abstract. Problems of conceptual uncertainty have been dealt with in theories of formal logic. Such theories try to accommodate vagueness in two main ways. One is fuzzy logic that introduces degrees of truth. The other way of accommodating formal logic to vagueness is super valuations and its descendants. This paper studies a more inclusive class of reasoning support than formal logic. In the present approach, conceptual uncertainty, including vagueness is represented as higher order uncertainty. A taxonomy of epistemic and conceptual uncertainty is provided. Finally, implications of conceptual uncertainty for reasoning support systems are analyzed.

1 Introduction

The purpose of computer based support systems for reasoning such as argumentation, decision and negotiation is to facilitate reasoning by replacing mental operations with externalized procedures, operating on representations in a computer medium. The design and development of computer support for reasoning, therefore, presupposes that we can find suitable external representations and describe procedures for operating on them.

A problem common to formal logic and more generally applicable reasoning procedures is the occurrence of conceptual uncertainty involving vagueness, fuzziness, ambiguity or open texture (Zadeh 1965, Fine 1975, Shapiro 2006). For instance, vagueness gives rise to problems of the law of excluded middle and of sorites paradoxes that are well known (Rolf 1981, Williamson and Graff 2002). If every proposition is true or false, is it true or false that France is hexagonal? If 0 grains do not make a heap but 100.000 grains do, which is the number n such that n grains did not make a heap but n+1 did? Such questions remind us that we are thinking by means of a conceptual system whose borders are not definitely sealed.

This paper studies the question: "How can computerized support systems for reasoning be accommodated to conceptual uncertainty? It asserts (1) Conceptual uncertainty is important for various types of reasoning such as argumentation, decision making and negotiations. (2) In reasoning supporting systems, conceptual uncertainty is representable as a second order uncertainty, directed towards the first order representations of matters of fact or courses of action. (3) There are five types or dimensions of uncertainty, two of them epistemic, three conceptual. The lens model of judgment theory confirms the taxonomy. (4) Conceptual and epistemic

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 672–682, 2007.

uncertainty can clarify reasoning in judgment and decision making. (5) Computer tools supporting reasoning need to consider both epistemic and conceptual uncertainty. This implies a need for a multiplicity of computer support tools together with corresponding skills to handle them.

2 Conceptual Uncertainty in Reasoning and Decision Making

In processes of reasoning and argumentation processes, there can occur two kinds of uncertainty (Smithson 1989, 2004). One is epistemic uncertainty consisting in uncertainty about what the facts are and what follows from what we know. Epistemic uncertainty is often represented in the form of a probability distribution with well-defined properties. Epistemic uncertainty is due to incompleteness of knowledge, i.e. ignorance.

Conceptual uncertainty is due to incompleteness of meaning, or intention. Even if you knew every single fact about a place – climate, vegetation or geology – you might still be uncertain whether it is "suitable for hiking". The latter uncertainty can only be reduced if we clarify whose hiking we are discussing and the preferences and abilities of those persons.

Conceptual uncertainty affects the two major models of rationality in decision making, consequentialist rationality and deontological rationality. Consequentialists think of decisions as a selection of alternatives. In such decisions, there will be conceptual uncertainty pertaining to alternatives and to attributes. Which alternatives are there and which attributes should one evaluate?

In deontological rationality, decisions are based on general principles such as laws, regulations, policies or instructions. Legal decision making is typical. In such decisions, conceptual uncertainty concerns which principles to apply and how to apply them, e.g. in case of conflicts between principles.

3 Conceptual Uncertainty of Level n Is Represented at n+1

Consider the type theory of Russell and Whitehead (Whitehead and Russell 1910). I will show how conceptual uncertainty relating to a factual assertion of the first level can be represented at the second level. Consider the assertion:

> The solar system has nine planets.

This statement was held true from 1930 when Pluto was discovered up till August 2006. Recently, however, Pluto has been found in peculiar company. The Kuiper belt, of which Pluto is a part, contains some 100 000 other objects. Some of them have the size of Pluto and one is actually larger. Pluto is very different from the other eight planets of the solar system.

It is certain that Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, and Neptune are planets. But there has for some time been uncertainty in the community of astronomers whether the judgment:

> Pluto is a planet.

is true or false. The uncertainty could stem from uncertainty about facts about Pluto or uncertainty about the conceptual delimitation of planethood. The uncertainty about the conceptual delimitation could not be resolved by knowledge about what the world is like.

In 2006, astronomers faced two possible delimitations of “planet”. One conceptual decision might be to define “planet” is a way that excludes Pluto and the rest of the objects in the Kuiper belt. Another conceptualization would include Pluto, but then it is problematic whether other objects of the Kuiper belt should be included as well.

The uncertainty about the planethood of Pluto could not be removed by a pure appeal to facts. It had to be removed in another way. The matter was settled by a decision at the XXVIth General Assembly of the International Astronomical Union (IAU) in August 2006 in Prague. The assembly had two options to define the concept of planet, one wider and one narrower. With a wider definition, based on a convention of 2005, the solar system would have contained twelve planets, including Pluto. But in 2006, the General Assembly of IAU settled for a narrower definition, stating that in the solar system, a planet is a celestial body that (1) is in orbit around the Sun, (2) has sufficient mass so that it assumes a hydrostatic equilibrium (nearly round) shape, and (3) has "cleared the neighborhood" around its orbit.

Together with the presently known facts, this definition settles that Pluto is not a planet. The solar system now contains eight planets, in the sense of “planet” defined in 2006.

The settling of the question whether Pluto is a planet is not of the same kind as the removal of other astronomical uncertainties or ignorance. Consider procedures for removing epistemic uncertainty about planets. In removing epistemic uncertainty, observations and inference from facts is used. The concept of “planet” is held fixed or, at least, not consciously elaborated upon. The planet Neptune was discovered in 1846 as a result of mathematical prediction. Perturbations in the orbit of Uranus led astronomers to deduce Neptune's existence. Applying analogous methods, the astronomer Urbain Le Verrier in 1859 tried to deduce a planet “Vulcan” that would have caused perturbations in the orbit of Mercury. There were several reports about observations of “Vulcan” up to 1915, when Einstein successfully explained the apparent anomaly in Mercury's orbit. As Einstein’s explanation became accepted, the search for Vulcan was abandoned by astronomers.

Instead, in removing the conceptual uncertainty of planet in 2006, a conceptual decision was needed. This decision was not an arbitrary one. It would have implications for the whole system of astronomical knowledge about the solar system and its planets. One cannot decide to redefine the concept of planet wider so as to include Pluto without also getting extra, unwanted, planets. Furthermore, had the wider definition been taken, the solar system might have come to include 53 presently known bodies in the solar system and possibly some hundreds of similar objects, presently unknown (*Pluto* Wikipedia 2006).

In fact, the decision about the definition of “planet” has been contested. The U.S. state of New Mexico's House of Representatives passed a resolution declaring that Pluto will always be considered a planet while overhead of the state, with March 13th being known as "Pluto Planet Day". Local news report that the widow and daughter of Clyde Tombaugh, discoverer of Pluto in 1930, participated in a solemn ceremony to reinstate the planethood of Pluto.

The uncertainty about there being a planet "Vulcan" was an epistemic uncertainty. Knowledge about facts could eliminate the uncertainty, *given the then prevailing sense of "planet"*. The uncertainty about Pluto that was ended or postponed by the IAU in Prague in August 2006 could not be ended by an appeal to facts. A conceptual decision either reinstating the previous sense of "planet" or settling a new sense was necessary to eliminate the conceptual uncertainty relating to the planethood of Pluto.

The uncertainty relating to the existence of a planet "Vulcan" is a first order uncertainty. The uncertainty concerns whether the facts are such as to make the following statement true:

> There is a planet between Mercury and the Sun.

The uncertainty about the planethood of Pluto is a second order uncertainty. We can represent it as an uncertainty about which to choose of the following two ways of rendering the concept of planet.

1. The predicate "planet" expresses the properties: (A) is in orbit around the Sun, (B) has sufficient mass so that it assumes a hydrostatic equilibrium (nearly round) shape.
2. The predicate "planet" expresses the properties: (A) is in orbit around the Sun, (B) has sufficient mass so that it assumes a hydrostatic equilibrium (nearly round) shape, and (C) "cleared the neighborhood" around its orbit.

The uncertainty about the planethood of Pluto can be represented as an uncertainty between the choice of 1 or 2 as ways to fixate the content of the predicate "planet".

Let me sum up some of the features of elaborations of epistemic versus conceptual uncertainty. (1) In science, epistemic uncertainty is diminished or removed ultimately by observation or experiment concerning putative facts together with inference from such. (2) Consequently, when epistemic uncertainty is removed, one obtains first order knowledge about facts or the implications of such knowledge. (3) In science, conceptual uncertainty is removed by decisions about the conceptual or representational system. (4) In a rational enterprise, such as science, decisions settling conceptual uncertainty are not arbitrary. They are made with considerations about consequences for large parts of a representational system of knowledge. (5) Consequently, in science, elaborations of conceptual uncertainty involve metacognition, i.e. knowledge and procedures involving our own system of knowledge. Such metacognition involves procedures of the logical second order.

One can identify epistemic uncertainty about what we would need to establish in order to settle a matter *holding the present conceptual system fixed*. Or we can identify conceptual uncertainty about what we would need to establish in order to settle a matter relating to *changes in our conceptual system*, holding facts fixed. Epistemic and conceptual uncertainty are abstract dimensions – one might need to elaborate on both dimensions in a particular case. This distinction – and the facts it relies on – seem to be in conflict with attempts to model vagueness as a kind of epistemic uncertainty. (cf Williamson 1994)

Similar considerations apply to normative systems such as law, regulations, policy, strategy or valuation. The two types of sources for uncertainty apply not only to cognition but also to values and norms. Administrative terminology draws on

compromises between science, folk wisdom, supposedly moral and religious commands, legal considerations, and political expediency. Conceptual elaborations might be called for in order to treat proper things in a proper manner. An example is the legal concept of death. When medical technology enabled hospitals to maintain a person's bodily functions in spite of serious brain damage, legal decisions fixated the concept of death.

Can we generalize our analysis about conceptual uncertainty? Conceptual uncertainty relating to choice is also representable at the second order. A consequentialist decision maker can be uncertain about the set of alternatives, the set of attributes and the methods for selecting an alternative, e.g. by compensatory, trade off methods or by non-compensatory methods, e.g. by some lexicographic method. A deontologist decision maker can be uncertain about which principles to select for application. Thus, conceptual uncertainty for a consequentialist and for a deontologist can be represented as something that needs to be settled by second order considerations.

4 A Taxonomy of Epistemic and Conceptual Uncertainty

We have claimed that epistemic uncertainty is absence of factual knowledge, i.e. knowledge of the first order, that would settle a question such as there being a planet "Vulcan" between Mercury and the Sun. We have claimed that conceptual uncertainty is the absence of knowledge of higher order that would settle a question such as Pluto being a planet. Such knowledge would need to rely on a fixation of the meaning of the predicate "planet".

Can the distinction we introduced be generalized to a theory of a more general distinction between kinds of uncertainty? I claim it can. There are, it seems, the following dimensions involved in solving any problem:

A. What is the problem? Identify the target problem, i.e. find a formulation of a question that covers the desire to make up one's mind. The question can be factual, mathematical, volitional, or preferential.
B. On which factors does the problem depend? Identify the factors that potentially bear sufficient relevance to the target problem. This can be an unstructured list of factors that might contribute to settling the question expressing the target problem.
C. What is the form of dependency? Identify the general form of influence that these factors bear to the target problem. This can, for instance, be an equational system, decomposing the dependency of the target vector into a set of equations with parameters and variables.
D. What is the strength of dependency? Identify the direction and strength by which the factors contribute. If the representation is in the form of an equational system, this aspect involves settling the parameters occurring in the equations.
E. What are the facts on which the solution to the problem rests? Identify the values of the variables expressing the factors of dependence.

Of these five dimensions of uncertainty, the first three A–C involve uncertainty of higher order and D and E involve uncertainty about facts, i.e. uncertainty of first order.

5 Arguments for the Taxonomy

Why these five and not three or seven dimensions of uncertainty? It is connected to the way that modeling can provide answers to problems. One needs to articulate the problem (A), the factors (B) and dependencies (C) that may be of importance. Finally, facts about the world and about preferences or norms of decision makers will settle the question, i.e. (D) and (E). The dimensions A–E are what it takes to settle a problem (cf. Nickles 1981).

The taxonomy clarifies the distinction between well structured problems and ill structured problems. Well structured problems are problems where A-C are settled while D and E need to be settled and there are known methods for settling D and E (Buckingham Shum 2003). Ill structured problems are open in some of the respects A, B or C. Those problems cannot be solved merely by an appeal to facts like D and E can.

The taxonomy fits the lens model of judgment theory. The model studies a subject utilizing certain cues to pass judgment on an object or a state. The cues can bear more or less *ecological validity* as indicators of the object/state. The subject may *utilize* these cues better or worse to achieve *accuracy* in her judgment. The accuracy can be represented as the product of ecological validity and cue utilization. The lens model has been applied to several types of judgment, from perceptual judgments via prognoses and predictions about the future (Payne 1993, Cooksey 1996, Hastie and Dawes 2001, Stewart 2000). The lens model can be used both descriptively and normatively. The cue utilization describes which weights a human judge or group of such lay to certain cues or indicators. If those weights are assigned in accordance with the ecological validity, one can achieve maximal accuracy in one's judgment, i.e. maximal rate of agreement with the real object or state about which one passes judgment.

The lens model represents a decomposition of our knowledge about facts and relations. The model can be expressed in two equations. One equation represents the ecological validity of the cues, i.e. how well knowledge of the cues would enable a judge to pass veridical judgment on the state of the object. The other equation represents the actual utilization of those cues by a human judge (or group of such).

Interestingly, these two equations can normally be represented by two linear functions. The cues that indicate the state of the object can be measured on scales that often permit a regression analysis, resulting in a linear function. Often, such a linear function represents the maximal accuracy that any human judge can reach (Dawes and Corrigan 1974).

A specific lens model, tied to an expert or group of experts forming judgment can be realized via the five dimensions of fixation. Any lens model will form a set of equations. Therefore, the five dimensions automatically apply.

6 Applications of the Taxonomy

Below, I will show the applicability of the taxonomy of conceptual uncertainty. A first way in which conceptual uncertainty enters into decision making is via the articulation of the target problem. For instance, one may ask whether the target factor can be spatially, temporally or causally subdivided. In an evaluation of Swedish measures taken against eutrophication in the Baltic, an international expert committee divided the target area into: (1) The Swedish east coast, (2) The open Baltic proper, (3) The Swedish west coast, (4) The Bothnian Bay, and (5) The Bothnian Sea (*Eutrophication of Swedish seas* 2006).

The division made is spatially and causally based. The Bothnian Bay and the Bothnian Sea do not seem to be affected of eutrophication. The mechanisms producing eutrophication and its consequences are different at the Swedish west coast were salt water prevents cyanobacterial blooms. A mixed set of measures is recommended and differences of causal mechanisms make different measures applicable at different places.

Sometimes, the articulation of the target problem can be facilitated via an articulation of the underlying qualities of ideal and of worst-case alternatives (Hammond et al. 1999). By so doing, one can come to an understanding of the dimensions underlying the decision at hand. Consider, for instance, eutrophication of the Baltic, a development everyone considers undesirable. But what is there about it that makes it undesirable: the deterioration of incomes from fishing, the massive algal blooms in the coastal regions or massive loss of oxygen-dependent life in the Baltic? By spelling out such underlying dimensions of the problem, one can, possibly, discover various sets of measures, relevant in different time spans.

A second way in which conceptual uncertainty enters is via the factors relevant to the target problem. One way to bring out factors of relevance uses top down procedures starting with definitions and other ways of making complex factors explicit. For instance, eutrophication is a concept that can be defined in a number of ways:

Eutrophication, however, is a condition in an aquatic ecosystem where high nutrient concentrations stimulate the growth of algae, which leads to imbalanced functioning of the system, such as: (1) intense algal growth: excess of filamentous algae and phytoplankton blooms; (2) production of excess organic matter; (3) increase in oxygen consumption; (4) oxygen depletion with recurrent internal loading of nutrients; and (5) death of benthic organisms, including fish (HELCOM Stakeholder Conference on the Baltic Sea Action Plan 2006*).*

Eutrophication is a process whereby water bodies, such as lakes, estuaries, or slow-moving streams receive excess nutrients that stimulate excessive plant growth (algae, periphyton attached algae, and nuisance plants weeds). This enhanced plant growth, often called an algal bloom, reduces dissolved oxygen in the water when dead plant material decomposes and can cause other organisms to die (*Eutrophication* US Geological Survey 2006).

Difference in wording may direct focus towards different aspects and point towards various solutions to a problem. We can compare the two underlying definitions:

	HELCOM	U.S.G.S.
What kind of things can undergo eutrophication?	an aquatic ecosystem	water bodies
What is the ontology of eutrophication?	a condition with certain consequences	process of delivery plus growth
What characterizes the entity?	high concentrations	excess nutrients

Fig. 1. Comparison between two definitions of *Eutrophication,* HELCOM vs. U.S.G.S.

Eutrophication is in HELCOM something that befalls aquatic ecosystems and such systems are not strictly identifiable with water bodies as U.S.G.S does. An ecosystem is causally delimited while a water body is spatially and temporally delimited. Concentrations of nutrients as *high* or in *excess* both involve comparisons but perhaps with different units. A concentration can be high without being in excess and conversely. While HELCOM focuses on a condition with its effects, U.S.G.S. focuses on a causal process of nutrient delivery and growth effect.

The slightly different definitions may open or close different possible activities to counter eutrophication. From the definitions, it would seem that HELCOM is committed to countermeasures for bringing concentrations down and preserving ecosystems, while U.S.G.S. might operate by introducing ecosystems consuming nutrients.

A third way in which conceptual uncertainty enters into decision making is via the general structure of dependence. As pointed out by Herbert Simon (Simon 1996, Chap. 8), our analysis and insight into complex systems rests on the assumption that they are nearly hierarchically decomposable. Most of his claims can be read as an epistemic or perhaps pragmatic claim about useful simplifications in causal system modeling (Agre 2003). For instance, in one example, he considers causal decomposability of the eutrophication process of Lake Erie. One way of representing eutrophication would be to build a complex simulation of interaction between grids in the lake, using equations to represent phosphate production and phosphate usage in each cell of the grid. Massive amounts of data about sewage plants and river mounds would be needed. Furthermore, predictions about growth of urban population, industry and agriculture would be needed in such a conception of modeling.

Instead, Simon suggests a temporal decomposition into three kinds of sub processes, one of long term change, one of short term change and one of intermediate term change. The long-term change is, Simon claims, gradual and hardly noticeable during human life spans. It is not needed in the model. The fastest process is the eutrophication. Therefore, Simon suggests that modeling should disregard other dynamic aspects underlying the whole process that are ill understood. The resulting, simplified model focuses on the relation between phosphate input and eutrophication. Finally, he notes that phosphate levels in the parts of the lake are strongly intercorrelated, so models can disregard differences between grids. Furthermore, human activities related to sewage treatment are too gross to be finely modeled. Simon concludes that more finely granulated models are superfluous to "social purposes" (Simon 1997 p. 109–111).

In Simon's example of a modeling process, decomposition serves the purpose of simplification of prediction with reasonable precision. "Our concern is not to forecast the future but (a) to understand the consequences of alternative possible futures, and (b) to understand which of these possible futures is associated with particular strategies or policy measures" (ib. p. 112). More interesting than prediction is sensitivity and an understanding of which the key variables are and what policy variables can have an effect, Simon holds.

As Simon indicates, the dependencies could be rendered in a modeling process with far more detail and with the ambition to predict. The general structure of dependence that Simon suggests has low granularity. Simon defends the lack of detail. We are ignorant about facts and causal relations, which make more fine-grained predictions impossible. Hence, first order uncertainty about facts interacts with second order uncertainties and second order decisions about the granularity of the representing system.

7 Computerized Tools for Judgment and Decisions

Do the concepts epistemic and conceptual uncertainty with our taxonomy help develop reasoning skills and design support tools? If so, in what way?

First, a complete computer support for reasoning skills must enable the representation and elaboration not only of epistemic uncertainty but also of conceptual uncertainty. Typically, decision support systems deal with the reduction of epistemic uncertainty only. Procedures for elaborating on epistemic support do not, by themselves, contribute to elaboration of conceptual uncertainty. But conceptual uncertainty is pervasive in decision making processes. We have already mentioned the conceptualization and selection of alternatives and attributes necessary in decision making.

Second, conceptual uncertainty is of a higher order than an uncertainty about facts. The representation of such uncertainty can, in many cases, be performed via multiple representations, e.g. a multitude of decision trees or of a multitude of delimitations of an ambiguously described factor or a vaguely indicated relation. This feature of uncertainty is compatible with super valuation approaches that represent vagueness or ambiguity as sets of precise models. It is doubtful whether these considerations are compatible with fuzzy logic, fuzzy set theory or other one-dimensional ways of representing vagueness. An uncertainty concerning the definitions of "planet" fits super valuation approaches but not a graded approach between two extremes, typical of fuzzy logic approaches.

Third, reasoning can involve rotations between procedures for elaborations on epistemic uncertainty and procedures for elaborating on conceptual uncertainty. For instance, a district attorney can be in doubt whether to press a charge of murder, manslaughter or severe assault. The burden of proof for the respective charges will differ. There is conceptual uncertainty about how to describe the deed together with epistemic uncertainty about what the facts of the case really are. Interaction between kinds of uncertainty obstructs sequential procedures for such rotation (But cf. Thagard 1992).

Fourth, in reasoning with conceptual uncertainty, one needs to be able to elaborate on multiple representations simultaneously. A problem can be symbolized in various ways and a decision maker can be uncertain about the best ways to construct or select a representation. Especially, multiple parties in a reasoning process can opt for different representations that, nevertheless, make competing claims about being the best ways to represent the (unarticulated) problem at hand. Therefore, a computer based reasoning supporting system should allow comparison between various representations (Rolf 2006a, 2006b).

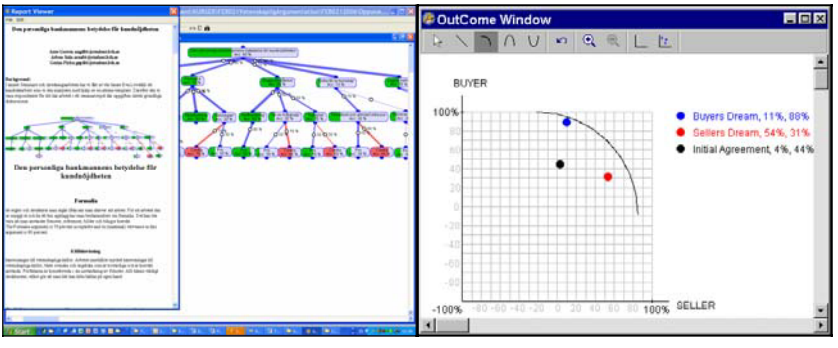


Fig. 2. Athena Standard (left), showing tree graph with report viewer presenting text output. Athena Negotiator (right), showing outcome diagram for two-party negotiation (Rolf 2002).

Human reasoning needs to be supported by multiple tools, providing multiple representations in order to represent conceptual uncertainty.

Finally, a computer-based support system will function as a tool in many respects. Each tool is part of a toolbox. There is no universal tool for all kinds of reasoning support. The various subtypes of uncertainty may have to be elaborated upon with different tools. The use of each tool needs competence, i.e. procedural knowledge. The use of the whole toolbox involves an overview of the purposes to which each tool can be put. When a designer introduces a software package for reasoning support, therefore, it is desirable to bear in mind that someone needs to supply heuristics for applying each of the tools as well as the whole toolbox.

Acknowledgments. This work has been granted support from the Swedish Environmental Protection Agency.

References

Agre, P.E.: Architecture of complexity. Journal of the Learning Sciences 12(3), 413–426 (2003)

Buckingham Shum, S.: The roots of computer supported argument visualization. In: Kirschner, P., et al. (eds.) Visualizing argumentation: software tools for collaborative and educational sense-making, Springer, London (2003)

Cooksey, R.W.: Judgment analysis. Academic Press, San Diego (1996)

Dawes, R., Corrigan, B.: Linear models in decision making. Psychological Bulletin 81(2), 95–106 (1974)
Eutrophication. U.S. Geological Survey, U.S.G.S (2006), http://toxics.usgs.gov/definitions/eutrophication.html
Eutrophication of Swedish seas. Report 5509. The Swedish Environmental Protection Agency, Stockholm (2006)
Fine, K.: Vagueness, truth and logic. Synthese 30, 265–300 (1975)
Graff, D., Williamson, T.: Vagueness. Ashgate-Dartmouth, Hants (2002)
Hammond, J.S., et al.: Smart choices. Harvard Business School Press (1999)
Hastie, R., Dawes, R.M.: Rational choice in an uncertain world. Sage Publications, Thousand Oaks (2001)
HELCOM. Stakeholder Conference on the Baltic Sea Action Plan, Helsinki, Finland (2006)
Nickles, T.: What is a problem that we may solve it? Synthese 47, 85–118 (1981)
Payne, W., et al.: The adaptive decision maker. Cambridge Univ. Press, Cambridge (1993)
Rolf, B.: Topics on vagueness. Studentlitteratur, Lund, Sweden (1981)
Rolf, B.: Athena software (2002), http://www.athenasoft.org
Rolf, B.: Decision support tools and two types of uncertainty reduction. In: Emmelin, L. (ed.) Effective environmental assessment tools. Blekinge Tekniska Högskola, Karlskrona, Sweden, Report 1, pp. 134–157 (2006a)
Rolf, B.: Logic software for apprenticeship in rough reasoning. In: Proceedings SICTTL. Univ. of Salamanca, Salamanca, Spain (2006b)
Shapiro, S.: Vagueness in context. Clarendon Press, Oxford (2006)
Simon, H.A.: The sciences of the artificial, 3rd edn. MIT Press, Cambridge (1996)
Simon, H.A.: Models of bounded rationality, vol. 3. The MIT Press, Cambridge (1997)
Smithson, M.: Ignorance and uncertainty. Springer, New York (1989)
Smithson, M.: Ignorance and uncertainty. In: Causality, uncertainty & ignorance. 3rd International Summer School, University of Konstanz, Germany (2004), http://www.uni-konstanz.de/ppm/summerschool2004/program.htm#smithson
Stewart, T.R.: Uncertainty, judgment, and error in prediction. In: Sarewitz, D., et al. (eds.) Prediction: science, decision making, and the future of nature, Island Press, Washington (2000)
Thagard, P.: Conceptual revolutions. Princeton University Press, Princeton (1992)
Whitehead, N.A., Russell, B.: Principia mathematica. Cambridge Univ. Press, London (1910)
Wikipedia.: Pluto (2006), www.wikipedia.org
Williamson, T.: Vagueness, Routledge, London (1994)
Zadeh, L.: Fuzzy sets. Information and Control 8, 338–353 (1965)

Reasoning with an Incomplete Information Exchange Policy

Laurence Cholvy[1] and Stéphanie Roussel[1,2]

[1] ONERA Centre de Toulouse
2 avenue Edouard Belin
31055 Toulouse, France
[2] SUPAERO
10 avenue Edouard Belin,
31055 Toulouse, France

Abstract. In this paper, we deal with information exchange policies that may exist in multi-agent systems in order to regulate exchanges of information between agents. More precisely, we discuss two properties of information exchange policies, that is the consistency and the completeness. After having defined what consistency and completeness mean for such policies, we propose two methods to deal with incomplete policies.

Keywords: completeness, information exchange policy, multi-agent system.

1 Introduction

Multi-agent systems provide an interesting framework for modelling systems in which some entities (atomic entities or complex ones) cooperate in order to fulfill a common task or to achieve a common goal. In order to cooperate efficiently, the entities, now called agents, have to exchange information, in particular in order to have a common view of the environment and a common understanding of the current situation.

In many systems, exchanges of information are not constrained and agents may exchange any information they want to anybody. At the opposite, in many other systems, information exchanges are ruled by a policy, in particular in order to satisfy some security constraints, like confidentiality, or efficiency constraints (broadcasting or peer-to-peer communication of relevant information). The so-called "Systems of Systems" in defense area or in civil security area [1] are instances of such multi-agent systems as well as any organisation of people and means like companies. These systems have in common that they are made of systems (human or not, atomic or not) which are geographically distributed, independently managed and which have to share information in a risky environment so that information exchanges between these systems must be compliant with a policy.

This present work deals with this last kind of systems. The illustrative example we will take all along the paper is the example of a hierarchical company with a boss and employees who exchange information relative to the materials used

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 683–694, 2007.

in the company. These exchanges must agree with a policy which, for instance, imposes the diffusion of pertinent and useful information as soon as possible, while respecting confidentiality restrictions.

An *information exchange policy* can then be seen as a regulation the agents must satisfy and which specifies the information exchanges which are obligatory, forbidden or permitted and under which conditions. But, in order to be useful, such a policy, as any other regulation, must satisfy several properties and in particular, it must be *consistent* and *complete*.

According to [2] which studies confidentiality policies, consistency allows to avoid cases when the user has both the permission and the prohibition to know something. More generally, according to [3] and [4], which study consistency of general kind of regulations, consistency of regulation does not come to classical consistency of a set of formulas. According to this work, a regulation is consistent if there exists no possible situation in which it leads an agent to *normative contradictions* or *dilemmas* also called in [15] *contradictory conflicts* (a given behaviour is prescribed and not prescribed, or prohibited and not prohibited) and *contrary conflicts* (a given behaviour is prescribed and prohibited). Following this definition, consistency of security policies has then been be studied in [5].

If consistency of policies is a notion that has been rather well studied, completeness has, at the opposite, received much less attention. [2] proposes a definition of completeness between two confidentiality policies (for each piece of information, the user must have either the permission to know it or the prohibition to know it), definition which has been adapted in [7] for multilevel security policies.

Recently, focusing on information exchange policies, a definition of consistency and a definition of completeness have been given in [6]. These definitions have constituted a starting point for the present work and have been refined.

This paper is organised as follows.

Section 2 presents the logical formalism used to express information exchange policies, the definition of consistency of such policies as well as the definition we give of completeness. Section 3 focuses on the problem of reasoning with an incomplete policy. Following the approach that has led to the CWA (Closed World Assumption) in Database area [14], we will present some rules of completion that can be used in order to complete an incomplete policy. Then, following [12], we will define some default rules and we will prove the equivalence between these two solutions. Section 4 is devoted to a discussion and extensions of this work will be mentionned.

2 Information Exchange Policies

2.1 Preliminaries

We use the framework defined in [6] to represent a sharing policy. This logical framework, L, is based upon a typed first order logic[1]. The alphabet of L will

[1] We use a first order logic instead of a modal deontic logic mainly because imbricating deontic modalities is not needed here. Furthermore, this allows us to use the results on policies consistency provided in [3].

be based on four distinct groups of symbols: constant symbols, variable symbols, predicate symbols and function symbols. As we want to type the language, we will distinguish different groups of symbols among those four categories.

Definition 1. *We distinguish three sets of constants: ag-constants (constants for agents), i-constants (constants for pieces of information), o-constants (other constants) and we distinguish three sets of variables: ag-variables (variables for agents), i-variables (variables for pieces of information), o-variables (other variables).*

Definition 2. *Predicate symbols are:*

- *D-predicates: unary predicates O, P, F and T (meaning respectively Obligatory, Permitted, Forbidden, Tolerated).*
- *P-predicates: predicates used to express any kind of property on pieces of information, agents, etc.*

Definition 3. *Functions symbols are:*

- *i-functions: used to represent properties about the pieces of information.*
- *not(.): unary-function used to represent object level negation.*
- *tell(.,.,.): function with three arguments representing the action of telling a piece of information. $tell(x, y, i)$ represents the event created by an agent x making the action of telling y a piece of information i.*

Definition 4. *Terms are defined the following way :*

- *ag-term : ag-constant or ag-variable*
- *i-term : i-constants and i-variables are i-terms. If f is an i-function and $i_1, ...i_n$ are i-terms then $f(i_1, ...i_n)$ is an i-term.*
- *d-term : If x and y are ag-terms and i is an i-term then $tell(x, y, i)$ is a d-term. Moreover, if d is a d-term then $not(d)$ is a d-term too.*
- *o-term : o-constant or o-variable*

Definition 5. *Formulas of L are defined recursively as follows:*

- *Let d be a d-term. Then $O(d)$, $P(d)$, $F(d)$ and $T(d)$ are D-literals and formulas of L.*
- *If $t_1, ...t_n$ are terms (other than d-terms) and P a P-predicate then $P(t_1, ..., t_n)$ is a P-literal and a formula of L.*
- *Let F_1 and F_2 be formulas of L and x be a variable. Then $\neg F_1$, $F_1 \wedge F_2$, $F_1 \vee F_2$, $\forall x\ F_1$, $\exists x\ F_1$, $F_1 \rightarrow F_2$ and $F_1 \leftrightarrow F_2$ are formulas of L.*

Example 1. We introduce here an example, that will be developped all along the paper. Let us consider the following logical language L: $a, b, c, Boss$ and $Employee$ are ag-constants and x and y are ag-variables. We can do the same for i-terms, etc. $Role(.,.)$ is a P-predicate. $Role(a, Boss)$ means that agent x plays the role $Boss$. $Topic(.,.)$ is a P-predicate. $Topic(i_1, ExpRisk)$ means that the piece of information i_1 deals with topic $ExpRisk$ (standing for Explosion Risk). $Agent(.)$ is a P-predicate. $Agent(b)$ means that b is an agent. $Receive(a, i_1)$ is a L-literal meaning that agent a receives the piece of information i_1. $O(tell(x, y, i))$ is a D-literal meaning that agent x is obligated to tell agent y the piece of information i.

2.2 Information Exchange Policies

In this section, we define rules for an information sharing policy, within the above logical language.

Definition 6. *An information sharing policy is a set of formulas of L which are conjunction of clauses* $l_1 \vee l_2 \vee ... \vee l_n$ *such that:*

- l_n *is the only positive literal and is a D-literal,*
- $\forall i \in \{1, ..., n-1\}$, l_i *is a negative L-literal, P-literal or D-literal,*
- *if* x *is a variable in* l_n*, then* $\exists i \in \{1, ..., n-1\}$ *such that* l_i *is a negative literal and contains the variable* x*.*

Example 2. The rule *"If a boss receives a piece of information dealing with the topic equipment checking, then it's forbidden for him to say it to his employees"* is expressed with the following formula:

$$(R_0) \quad \forall(x, y, i) \quad Role(x, Boss) \wedge Role(y, Employee)$$

$$\wedge Receive(x, i) \wedge Topic(i, EqtChk) \rightarrow F(tell(x, y, i))$$

2.3 Consistency and Completeness of Policies

We note $\mathcal{A}$ the following set of axioms:

$$(Ax1) \quad \forall x \quad P(x) \leftrightarrow \neg O(not(x)) \qquad (Ax2) \quad \forall x \quad F(x) \leftrightarrow O(not(x))$$

$$(Ax3) \quad \forall x \quad T(x) \leftrightarrow P(x) \wedge P(not(x)) \qquad (D) \quad \forall x \quad O(not(x)) \rightarrow \neg O(x)$$

$$(NO) \quad \forall x \quad O(not^{2n}(x)) \leftrightarrow O(x) \qquad (NP) \quad \forall x \quad P(not^{2n}(x)) \leftrightarrow P(x)$$

$$(NF) \quad \forall x \quad F(not^{2n}(x)) \leftrightarrow F(x)$$

NOTATION : Let $A_1, A_2,$ and A_3 be formulas of L. We will note:
$A_1 \otimes A_2$ instead of $(A_1 \vee A_2) \wedge \neg(A_1 \wedge A_2)$ and
$A_1 \otimes A_2 \otimes A_3$ instead of $(A_1 \vee A_2 \vee A_3) \wedge \neg(A_1 \wedge A_2) \wedge \neg(A_2 \wedge A_3) \wedge \neg(A_1 \wedge A_3)$.
This notation means that one and only one of the formulas A_i is true.

Theorem 1. $\forall d$ *d-term* $\mathcal{A} \models O(d) \otimes T(d) \otimes F(d)$

Proof. $\forall x, \neg T(x) \leftrightarrow_{Ax3} \neg P(x) \vee \neg P(not(x))$. Yet, $\neg P(x) \leftrightarrow_{Ax1} O(not(x)) \leftrightarrow_{Ax2} F(x)$ and $\neg P(not(x)) \leftrightarrow_{Ax1} O(not^2(x)) \leftrightarrow_{NO} O(x)$. Thus $\mathcal{A} \models \neg T(x) \leftrightarrow O(x) \vee F(x)$ and $\mathcal{A} \models O(x) \vee T(x) \vee F(x)$. Then, we have $\mathcal{A} \models \neg(O(d) \wedge T(d))$, $\mathcal{A} \models \neg(O(d) \wedge F(d))$ and $\mathcal{A} \models \neg(T(d) \wedge F(d))$.

Definition 7. *A formula or a set of formulas S is complete if and only if, for all P-literal l, we have:* $S \models l$ *or* $S \models \neg l$*.*

Definition 8. *A state of the world or a world, W, is a set of atomic formulas of L without D-literals. If this set is complete, we speak about a complete world.*

Let us note *Dom* the set of constraints that are supposed to be true in all worlds.

Definition 9. *Let $\mathcal{P}$ be a policy defined as a set of formulas of L, and W a complete world ruled by $\mathcal{P}$. $\mathcal{P}$ is consistent in W (according to Dom) if and only if $(W \wedge Dom \wedge P \wedge \mathcal{A})$ is consistent.*

Example 3. We take $Dom = \{\}$; Let us consider the following world W_0[2]:

$$W_0 = \{Agent(a), Agent(b), Role(a, Boss), Role(b, Employee)$$

$$Theme(i_1, EqtChk), Theme(i_2, ExpRisk), Receive(a, i_2)\}.$$

Let $\mathcal{P}_0$ be a policy containing one rule which is the rule (R_0). $(W_0, Dom, \mathcal{P}_0, \mathcal{A})$ is consistent. Thus, $\mathcal{P}_0$ is consistent in W_0.

Definition 10. *Let $\mathcal{P}$ be a policy. $\mathcal{P}$ is consistent (according to Dom) if and only if there is no set of formulas f of L without D-literal such that $(f \wedge Dom)$ is consistent and $(\mathcal{P} \wedge \mathcal{A} \wedge f \wedge Dom)$ is inconsistent.*

Proposition 1. *$\mathcal{P}$ is consistent (according to Dom) if and only if for all complete world W, $\mathcal{P}$ is consistent in W.*

Proof. This can be proved by using contraposition.

Example 4. Let (R_1) be the following rule: *"When an employee receives any piece of information about equipment check, it's tolerated for him to tell it to another employee"*. (R_1) can be formalized in the following way:

$$(R_1) \quad \forall(x, y, i) \quad Role(x, Employee) \wedge Role(y, Employee) \wedge \neg(x = y)$$

$$\wedge Receive(x, i) \wedge Topic(i, EqtChk) \rightarrow T(tell(x, y, i))$$

Let us consider the policy $\mathcal{P}_1$ containing rules (R_0) and (R_1). If we take $f = Role(a, Employee) \wedge Role(a, Boss)$, then (R_0) allows us to infer $F(tell(a, y, i_2))$ and (R_1) to infer $T(tell(a, y, i_2))$. Thus, we have a contradiction and $\mathcal{P}_1$ is not consistent in W_0 so not globally consistent.

Intuitively, for a given world, a policy is complete if it allows to deduce the behaviour that any agent should have, according to any piece of information and according to any other agent he could tell this piece of information. It could be obligatory, forbidden or tolerated for the agent to say the piece of information to the other agent.

Definition 11. *Let $\mathcal{P}$ be a policy and W a complete world ruled by $\mathcal{P}$. $\mathcal{P}$ is complete for $\models$ in W if and only if, for all $X = (x, y, i)$*

$$If\ W \models Receive(x, i) \wedge Agent(y) \wedge \neg(x = y)\ Then$$

$$(\mathcal{P}, W, \mathcal{A} \models O(tell(X))\ or\ \mathcal{P}, W, \mathcal{A} \models F(tell(X))\ or\ \mathcal{P}, W, \mathcal{A} \models T(tell(X))$$

[2] We will write in W only the positive literals for more readability. Each literal that is not explicitely written in W will be considered as negative.

This definition can be generalized and we can define a global completeness.

Definition 12. *Let $\mathcal{P}$ be a policy. $\mathcal{P}$ is globally complete for $\models$ if and only if for all complete world W, $\mathcal{P}$ is complete for $\models$ in W.*

Example 5. We have $W_0 \models Receive(a, i_2) \wedge Agent(b) \wedge \neg(a = b)$ but $\mathcal{P}_0, W_0, \mathcal{A} \not\models O(tell(a, b, i_2))$ and $\mathcal{P}_0, W_0, \mathcal{A} \not\models T(tell(a, b, i_2))$ and $\mathcal{P}_0, W_0, \mathcal{A} \not\models F(tell(a, b, i_2))$. Thus, $\mathcal{P}_0$ is incomplete for $\models$.

Completeness is an important issue for a policy. For a given situation, without any behaviour stipulated, any behaviour could be observed and thus consequences could be quite important. With an incomplete policy, we could detect the "holes" of the policy and send them back to the policy designers so that they can correct them or we could detect the "holes" of the policy and allow for those holes some default rules that could be applied to correct them. The first solution could be quite irksome to be applied (the number of holes could be quite important and thus correct them one by one quite long). Then, we put in place the second solution.

3 Reasoning with Incomplete Policies

3.1 Completion Rules

In this paragraph, we present a solution which extends the CWA defined by Reiter to complete first order databases.

According to CWA, if the database is incomplete for a literal l (i.e l is not deduced in the database), then it can be assumed that its negation ($\neg l$) is deduced. This rule is motivated by the assumption that a database is used to represent the real world. Since in the real world, a fact is true or is false (i.e $l \otimes \neg l$ is a tautology in first order logic) then a database must deduce a fact or it negation.

Here, given a d-term l, we are not interested in its truth value but in the fact that a given policy deduces that it is obligatory, forbidden or tolerated. These three cases are the only ones because axioms $\mathcal{A}$ imply $O(l) \otimes F(l) \otimes T(l)$. Thus, if the policy is incomplete for a literal l (i.e it does not deduce neither $O(l)$ nor $F(l)$ nor $T(l)$) then it can only be completed by assuming that $O(l)$ can be deduced, or $P(l)$ or $F(l)$. This leads to the three completion rules which are described in the following.

Furthermore, in order to be as general as possible, we define parametrized completion rules so that the way of completing by $O(l)$, $P(l)$ or $F(l)$ may depend on some conditions. These conditions, denoted E_i in the following, will represent properties about agents (e.g, agents having a specific role), information (pieces of information dealing with a specific topic),etc.

Let $\mathcal{P}$ be a consistent policy and W be a complete world ruled by $\mathcal{P}$.

Notation. For more readabilty, we will write *"$\mathcal{P}, W$ incomplete for (x, y, i)"* instead of: $W \models Receive(x, i) \wedge Agent(y) \wedge \neg(x = y)$ and $\mathcal{P}, W, \mathcal{A} \not\models O(tell(x, y, i))$ and $\mathcal{P}, W, \mathcal{A} \not\models T(tell(x, y, i))$ and $\mathcal{P}, W, \mathcal{A} \not\models F(tell(x, y, i))$.

Let E_1, E_2 and E_3 be three formulas that depend on x and/or on y and/or on i. We will write X instead of (x, y, i)

The three inference rules are:

$$(R_{E_1}) \qquad \frac{\mathcal{P}, W \; incomplete \; for \; X, \quad W \models E_1(X)}{F(tell(X))}$$

$$(R_{E_2}) \qquad \frac{\mathcal{P}, W \; incomplete \; for \; X, \quad W \models E_2(X)}{T(tell(X))}$$

$$(R_{E_3}) \qquad \frac{\mathcal{P}, W \; incomplete \; for \; X, \quad W \models E_3(X)}{O(tell(X)}$$

We can complete an incomplete policy so that it is obligatory (R_{E_1}), forbidden (R_{E_2}) or tolerated (R_{E_3}) for an agent to tell another agent a piece of information, according to those three elements. We define here a **new inference** that we will note $\models_*$. Rules of inference for $\models_*$ are the same as for $\models$ but we add R_{E_1}, R_{E_2} and R_{E_3}.

The next step is to verify that the policy is complete and consistent with this new inference.

First of all, we have to extend the definition of completeness of a policy with the inference $\models_*$.

Definition 13. *Let $\mathcal{P}$ be a policy and W a complete world ruled by $\mathcal{P}$. $\mathcal{P}$ is complete for $\models_*$ in W if and only if for all $X = (x, y, i)$, we have:*

$$If \; W \models Receive(x, i) \wedge Agent(y) \wedge \neg(x = y) \; Then$$

$$(\mathcal{P}, W, \mathcal{A} \models_* O(tell(X)) \; or \; \mathcal{P}, W, \mathcal{A} \models_* T(tell(X)) \; or \; \mathcal{P}, W, \mathcal{A} \models_* F(tell(X)))$$

This definition can be generalized.

Definition 14. *Let $\mathcal{P}$ be a policy. $\mathcal{P}$ is globally complete if and only if for all complete world W, $\mathcal{P}$ is complete in W.*

Proposition 2. *Let $\mathcal{P}$ be a policy and W a complete world ruled by $\mathcal{P}$. $\mathcal{P}$ is complete for $\models_*$ in W if and only if*

$$\forall X = (x, y, i), \mathcal{P}, W \; incomplete \; for \; X \Rightarrow W \models E_1(X) \vee E_2(X) \vee E_3(X))$$

Proof. This can be proved by reasoning with contraposition.

Example 6. $E_1(x, y, i) = Topic(i, EqtCheck)$, $E_2(x, y, i) = False$, $E_3(x, y, i) = Topic(i, ExpRisk)$. We have $\mathcal{P}_0, W_0$ incomplete only for (a, b, i_2). We have $W_0 \models E_3(a, b, i_2)$ so $W_0 \models (E_1(a, b, i_2) \vee E_2(a, b, i_2) \vee E_3(a, b, i_2))$. Then the poliy $\mathcal{P}_0$ is complete in W_0.

Then, we have to extend the definition of consistency for the new inference.

Definition 15. *Let W be a complete a world and $\mathcal{P}$ a policy that is consistent for $\models$ in W* [3]*. $\mathcal{P}$ is consistent for $\models_*$ in W (according to domain Dom) if and only if $W, Dom, \mathcal{P}, \mathcal{A}$ is consistent for $\models_*$ (i.e $W, Dom, \mathcal{P}, \mathcal{A} \not\models \bot$).*

Proposition 3. *A policy $\mathcal{P}$ that is complete for $\models_*$ in a complete world W is consistent for $\models_*$ in W (according to Dom) if and only if*

$$\forall X = (x, y, i) \quad \textit{If } \mathcal{P}, W \textit{ incomplete for } (x, y, i) \textit{ Then}$$

$$W \models \neg(E_1(X) \wedge E_2(X)) \wedge \neg(E_1(X) \wedge E_3(X)) \wedge \neg(E_2(X) \wedge E_3(X))$$

Proof. This can be proved by reasoning with contraposition.

Example 7. We take $E_1(x, y, i) = Topic(i, EqtChk)$, $E_2(x, y, i) = False$ and $E_3(x, y, i) = Topic(i, ExpRisk)$. We have verified that $\mathcal{P}_0$ is complete for $\models_*$ in W_0. $\mathcal{P}_0, W_0$ is incomplete for (a, b, i_2). For this triplet, we have $W_0 \models \neg(E_1(a, b, i_2) \wedge E_3(a, b, i_2))$. Thus, the policy $\mathcal{P}_0$ is consistent for $\models_*$ in W_0.

Corollary 1. *Let $\mathcal{P}$ be a policy and W a world ruled by $\mathcal{P}$. $\mathcal{P}$ is consistent and complete for $\models_*$ in W if and only if*

$$\forall X = (x, y, i) \textit{If } \mathcal{P}, W \textit{ incomplete for } (x, y, i) \textit{ Then } W \models E_1(X) \otimes E_2(X) \otimes E_3(X)$$

Definition 16. *A policy $\mathcal{P}$ is globally consistent for $\models_*$ (according to Dom) if and only if it is consistent for $\models_*$ in all complete world W where $W \wedge Dom$ is consistent.*

3.2 Default Rules

The three rules that we have just defined look similar to default logic defined by Reiter in [12]. The aim of this section is to develop this aspect. As a reminder to the default theory, one could read the chapter dedicated to default rules in [11]. Let $\mathcal{P}$ be a policy in a complete world W. We suppose that $\mathcal{P}$ is consistent in W for $\models$. Let W' be the set of formulas defined by $W' = \mathcal{P} \cup W \cup \mathcal{A}$. We define three default rules in W' and for that, we consider a triplet $X = (x, y, i)$.

$$(d_1)\ \frac{Receive(x, i) \wedge Agent(y) \wedge \neg(x = y) \wedge E_1(X) \ :\ F(tell(X))}{F(tell(X))}$$

$$(d_2)\ \frac{Receive(x, i) \wedge Agent(y) \wedge \neg(x = y) \wedge E_2(X) \ :\ T(tell(X))}{T(tell(X))}$$

$$(d_3)\ \frac{Receive(x, i) \wedge Agent(y) \wedge \neg(x = y) \wedge E_3(X) \ :\ O(tell(X))}{O(tell(X))}$$

The default rule d_1 can be read as following: *"If, in W', an agent x receives a piece of information i, if y is another agent, if $X = (x, y, i)$ are such that $E_1(X)$*

[3] It's not relevant to study a policy that is not consistent in W.

is true, and if it is consistent to suppose that it is fordidden for x to say i to y, then we consider that this fordidding is true in W'".

We note $D = \{d_1, d_2, d_3\}$. d_j ($j \in \{1, 2, 3\}$) is applicable if its prerequisite can be infered in W' and if the negation of its justification cannot be infered in W'. We consider now the theory $\Delta = (D, W')$ and we look at its possible extensions.

Proposition 4. *The default theory $\Delta = (D, W')$ has at least one consistent extension.*

Proof. $\Delta = (D, W')$ is a closed normal default theory so we can use Reiter's theorem (theorem 3.1 in [12]) that says that *"Every closed normal default theory has an extension"*. Thus, as W' is consistent, we can deduce that this extension is consistent.

Example 8. We build the three default rules with $E_1(x, y, i) = Topic(i, EqtChk)$, $E_2(x, y, i) = False$ and $E_3(x, y, i) = Topic(i, ExpRisk)$. We have:

$W'_0 \models Receive(a, i_2) \wedge Agent(b) \wedge \neg(a = b) \wedge E_3(a, b, i_2)$ and $W'_0 \not\models \neg O(tell(a, b, i_2))$. The default rule d_3 is then applicable for (a, b, i_2). An extension of $\Delta_0 = (D, W'_0)$ could be $E_{\Delta_0} = Th(W'_0) \cup \{O(tell(a, b, i_2)), P(tell(a, b, i_2))\}$.

We use here the universal inference for default rules : Let φ be a formula of L. $W \models^{UNI,D} \varphi$ if and only if φ belongs to every extension of (D, W).

Proposition 5. *Let $\mathcal{P}$ be a policy applied in a complete world W. We suppose that $\mathcal{P}$ is consistent in W.*
$\Delta = (D, W')$ has one and only one extension E_Δ if and only if

$$\forall X = (x, y, i) If \quad \mathcal{P}, W \text{ incomplete for } X \text{ Then}$$

$$W \models \neg(E_1(X) \wedge E_2(X)) \wedge \neg(E_1(X) \wedge E_3(X)) \wedge \neg(E_2(X) \wedge E_3(X)$$

Example 9. $\mathcal{P}, W$ is incomplete only for $X_0 = (a, b, i_2)$. We have $W \models \neg(E_1(X_0) \wedge E_2(X_0)) \wedge \neg(E_1(X_0) \wedge E_3(X_0)) \wedge \neg(E_2(X_0) \wedge E_3(X_0)$ then E_{Δ_0} is the only extension of Δ_0.

Definition 17. *Let $\mathcal{P}$ be a policy applied in a complete world W. $\mathcal{P}$ is consistent for Dom in W for $\models^{UNI,D}$ if and only if $(D, W' \cup Dom)$ has one and only one extension.*

Definition 18. *Let $\mathcal{P}$ be a policy applied in a complete world W. $\mathcal{P}$ is complete for $\models^{UNI,D}$ in W if and only if we have*

$$\forall X = (x, y, i) \text{ If } W \models Receive(x, i) \wedge Agent(y) \wedge \neg(x = y) \text{ Then}$$

$$(W' \models^{UNI,D} O(tell(X)) \text{ or } W' \models^{UNI,D} T(tell(X)) \text{ or } W' \models^{UNI,D} F(tell(X)))$$

Proposition 6. *Let $\mathcal{P}$ be a policy applied in a complete world W. $\mathcal{P}$ is consistent for Dom in W and is complete in W for $\models^{UNI,D}$ if and only if*

$$\forall X = (x, y, i) If \mathcal{P}, W \text{ incomplete for } X \text{ Then } W \models E_1(X) \otimes E_2(X) \otimes E_3(X)$$

Example 10. For $X_0 = (a, b, i_2)$, we have $W_0 \models E_1(X_0) \otimes E_2(X_0) \otimes E_3(X_0)$. P_0 is consistent and complete in W_0 pour $\models^{UNI,D}$.

3.3 Comparison

The two methods that we have just seen look very similar. The following proposition shows their relation.

Proposition 7. *Let W be a complete world and $\mathcal{P}$ a policy that rules W. For all $X = (x, y, i)$*

$$\begin{aligned} &\textit{If } W \models \textit{Receive}(x,i) \wedge \textit{Agent}(y) \wedge \neg(x = y) \textit{ Then} \\ (\ &\mathcal{P}, W, \mathcal{A} \models_* F(\textit{tell}(X)) \Leftrightarrow W' \models^{UNI,D} F(\textit{tell}(X)) \wedge \\ &\mathcal{P}, W, \mathcal{A} \models_* T(\textit{tell}(X)) \Leftrightarrow W' \models^{UNI,D} T(\textit{tell}(X)) \wedge \\ &\mathcal{P}, W, \mathcal{A} \models_* O(\textit{tell}(X)) \Leftrightarrow W' \models^{UNI,D} O(\textit{tell}(X))\ ) \end{aligned}$$

4 Discussion

After having given a logical framework and showed how to formalize a information exchange policy within this framework, we have reminded of a definition of consistency and we have defined what meant completeness for a policy. Therefore, the issue was to deal with incomplete policies. In solution to that, we have proposed two ways of completing a policy. One way is to use a new inference with three inference rules that can be applied for elements for which the policy is incomplete. The other one is to use the default theory and in particular three default rules that can be applied as soon as they are not in contradiction with what already exists. These default rules allow the construction of a new complete policy. After having completed policies, we can check that the result is still consistent. Finally, we prove that the methods results are equivalent. Indeed, in a given situation, as soon as an agent will receive a piece of information, the question will be to check if the policy deduces that it is obligatory, tolerated or forbidden for him to tell that information to another agent. If the anwser is negative, then the question will be to check which condition I_i is true. If E_1 (resp, E_2, E_3) is true, then the agent will deduce that it is forbidden (resp, tolerated, obligatory) to exchange this information. The condition on the I_i's ensures that the agent will deduce one of them.

This work could be extended in many directions.

First, we could extend it by adding the notion of time. As it is shown in [10], this issue is very important when speaking about obligations. In our case, the impact of time will be quite difficult to deal with. Actually, we will have to consider different times : time the piece of information is created, time it is received by an agent, time the obligation is created, time the agent makes the action of telling the piece of information, time the obligation lasts.

Secondly, we could focus our attention on the so-called *Receive* predicate and study its semantic in relation with the agent's belief base revision. Indeed, the obligations, prohibitions, tolerances expressed by the policy should not be triggered by the arrival of a new piece of information in the agent's base, but by the computation of the "new" beliefs (i.e the ones which belong to the difference between the base before and after the revision).

Besides, it must be noticed that some properties other than consistency and completeness could be studied. For instance, we could wonder if the notion of *correctness*, as introduced in [8], is pertinent in our case. In that paper, the authors introduce the notion of *correctness* of airport security regulation. In their context, there are different organizations which do not have the same hierarchy level but which create rules on the same topics (e.g, dangerous objects on board a aircraft). The higher the hierarchy level of the organization is, the more general the rule is. The lower hierarchy level organizations have to create sub-rules that once all checked will validate the general rules. *Correctness* ensures that the sub-rules fulfill the general rules.

Finally, this present work could be extented to any kind of regulations. Indeed, it must be noticed that the idea which underlines the notion of completeness studied here for information exchange policies could be used to characterize a kind of "local completeness" (completeness relatively to some particular situations) for any type of regulation. This notion of "local completeness" is rather similar to the one already introduced in Database area. Indeed, as mentionned in [13] and [9], some integrity constraints expressed on a database are in fact rules about what the database should know. For instance, considering a database of a multi-national company, the integrity constraint "any employee has got a phone number, a fax number or a mail adress" expresses in fact that, for any employee known by the database, the database knows its phone number, its fax number or its mail address. [4]. As first mentionned by Reiter [13], this integrity constraint expresses a kind of local completeness of the database. Defaults, as reiter defined them, can be used in order to complete such a database in case of incompleteness. For instance, one of the rules can be that if the database does contain any required information (no phone number, no fax number, no mail address) for a given employee but if the department that employee works in is known, then it can be assumed that its phone number is the phone number of its department.

Acknowledgements

The authors wish to thank the reviewers for their helpful criticism.

References

1. IEEE international conference on systems of systems engineering (2006)
2. Bieber, P., Cuppens, F.: Expression of confidentiality policies with deontic logic. In: Proceedings of the First Workshop on Deontic Logic and Computer Science (DEON'91) (1991)
3. Cholvy, L.: An application of SOL-deduction: checking regulation consistency. In: IJCAI'97 Poster Collection (1997)

[4] Notice that this does not prevent the fact that in the real world, an employee of the company has no telephone number, no fax number and no mail address.

4. Cholvy, L.: Checking regulation consistency by using SOL-resolution. In: International Conference on Artificial Intelligence and Law, pp. 73–79 (1999)
5. Cholvy, L., Cuppens, F.: Analysing consistency of security policies. In: IEEE Symposium on Security and Privacy (1997)
6. Cholvy, L., Garion, C., Saurel, C.: Modélisation de réglementations pour le partage d'informations dans un SMA. In: Modèles Formels de l'Interaction (2007)
7. Cuppens, F., Demolombe, R.: A modal logical framework for security policies. Lectures Notes in Artificial Intelligence, vol. 1325. Springer, Heidelberg (1997)
8. Delahaye, D., Etienne, J., Donzeau-Gouge, V.: Reasoning about airport security regulations using the focal environment. In: 2nd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (2006)
9. Demolombe, R.: Database validity and completeness: another approach and its formalisation in modal logic. In: Proc. IJCAI Workshop on Knowledge representation meets databases (1999)
10. Demolombe, R., Bretier, P., Louis, V.: Formalisation de l'obligation de faire avec délais. In: Proc. MFI'2005 (2005)
11. LEASOMBE. Logique des défauts. In: Raisonnement sur des informations incomplètes en intelligence artificielle. Teknea, Marseille (1989)
12. Reiter, R.: A logic for default reasoning. Artificial Intelligence 13(1,2) (1980)
13. Reiter, R.: What should a database know? J. Log. Program. 14(1-2), 127–153 (1992)
14. Reiter, R.: On closed world data bases. In: Nicolas, J.M., Gallaire, H., Minker, J. (eds.) Logic and Databases, Plenum Publications, New-York (1998)
15. Vranes, E.: The definition of "norm conflict" in international law and legal theory. The European Journal of International Law 17(2), 395–418 (2006)

Qualitative Constraint Enforcement in Advanced Policy Specification

Alessandra Mileo[1] and Torsten Schaub[2,⋆]

[1] Dipartimento di Informatica e Comunicazione, Universitá degli Studi di Milano-Bicocca — Lab. Nomadis, via Bicocca degli Arcimboldi 8, I–20126 Milano
mileo@dico.unimi.it
[2] Institut für Informatik, Universität Potsdam, Postfach 90 03 27, D–14439 Potsdam
torsten@cs.uni-potsdam.de

Abstract. We consider advanced policy description specifications in the context of Answer Set Programming (ASP). Motivated by our application scenario, we further extend an existing policy description language, so that it allows for expressing preferences among sets of objects. This is done by extending the concept of ordered disjunctions to cardinality constraints. We demonstrate that this extension is obtained by combining existing ASP techniques and show how it allows for handling advanced policy description specifications.

1 Introduction

The specification of policies and their enforcement plays a key role in advanced system environments, where a large variety of events, conditions and actions are to be executed and monitored. The development and analysis of a collection of such policies can be rather complex, in particular, in view of their overall consistency. To this end, a high-level policy description language called PDL has been developed by Chomicki, Lobo and Naqvi [1] in the context of Network management, through a mapping into Answer Set Programming (ASP;[2]).

A first extension of PDL lead to the description of PPDL language [4,3]. In PPDL a *policy* is a set of event-condition-action rules describing how events observed in a system, trigger actions to be executed, and a *consistency monitors* is a set of rules of the form:

$$\textbf{never } a_1 \times \ldots \times a_n \textbf{ if } C. \tag{1}$$

meaning that actions named $a_1, \ldots, a_n$ cannot be jointly executed. In case condition C holds and actions $a_1, \ldots, a_n$ have all been triggered by the policy application, a_1 should be preferably blocked, if this is not possible (i.e. a_1 must be performed), a_2 should be blocked, ..., then if all of $a_1, \ldots, a_{n-1}$ must be performed, then a_n must be blocked.

⋆ Affiliated with the School of Computing Science at Simon Fraser University, Burnaby, Canada.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 695–706, 2007.

A rule as in (1) is mapped into ASP through LPOD [5] encoding as follows:

$$\begin{array}{ll} block(a_1) \times \ldots \times block(a_n) & \leftarrow exec(a_1), \ldots, exec(a_n), C. \\ accept(A) & \leftarrow not\ block(A). \end{array} \quad (2)$$

where $block(a_i)$ indicates conflicting actions that have to be blocked, $exec(a_i)$ refers to actions triggered by the policy application, and $accept(a_i)$ tells us which actions can be executed without any constraint violation.

As illustrated in [4], the introduction of user-preferences in PPDL *monitor* rules enables users to tell the system in which order to enforce constraints on the execution of actions triggered by the policy.

From the viewpoint of policy enforcement, it is often the case that an ordering relation among users, resources and more generally, among objects on which actions have to be executed, is to be expressed on sets of entities having certain characteristics or being hierarchically organized.

As an example, consider the context of resource management: reliability of a resource could be influenced by its use and (dynamically determined) performance. Preference relation on actions involving resources has to be dynamic too.[1] Besides the dynamic nature of our logic-based approach, preferences on sets are much more intuitive than static classification of objects. Formal aspects related to the specification of *preferential monitors* in PPDL have been fully addressed in [4,3] by appeal to LPOD programs.

Another interesting aspect is related to expressing a further ordering among strategies (represented by monitor rules) for conflicts resolution.

We address these issues in the policy enforcement context by extending LPOD to allow for ordered disjunctions of cardinality constraints and we call this extension S-LPOD. Moreover, we consider the preference relation on LPOD rules introduced by Brewka et al. [6], by discussing some of its properties, and we apply it to S-LPOD rules, resulting in so-called SR-LPOD programs.

Given our application-oriented motivation, we tried to keep our formal development as conservative as possible in relying on existing approaches whenever feasible. Fortunately, this is achievable in straightforward way due to the compositional nature of many ASP extensions.

2 Background

To begin, we recall the basic definitions of Logic Program with Ordered Disjunction (LPOD), as given in [5] and [6]. For basic definitions in Answer Set Programming, we refer the reader to [2].

Given an alphabet $\mathcal{P}$ of propositional symbols, an LPOD-program is a finite set of LPOD-rules of the form

$$c_1 \times \ldots \times c_l \leftarrow a_1, \ldots, a_m, not\ b_1, \ldots, not\ b_n. \quad (3)$$

where each a_i, b_j, c_k is a *literal*, that is, an *atom* $p \in \mathcal{P}$ or its negation $\neg p$ for $0 \leq i \leq m$, $0 \leq j \leq n$, and $0 \leq k \leq l$ If $m = n = 0$, then (3) is a

[1] See Section 5 for further details related to this context.

fact. If $l = 1$, then we have a normal rule. If $l = 0$, then (3) is an integrity constraint. (cf. [2]) For a rule r as in (3), let $head(r) = \{c_1, \ldots, c_l\}$ be the *head* of r and $body(r) = \{a_1, \ldots, a_m, not\ b_1, \ldots, not\ b_n\}$ be the *body* of r; and let $body^+(r) = \{a_1, \ldots, a_m\}$ and $body^-(r) = \{b_1, \ldots, b_n\}$.

The "non-standard" part of such a rule is the *ordered disjunction* $c_1 \times \ldots \times c_l$ constituting its head. Given that the body literals are satisfied, its intuitive reading is:

- if possible c_1, but if c_1 is impossible, then c_2,
- ...,
- if all of $c_1, \ldots, c_{l-1}$ are impossible, then c_l.

Each c_k stands for a *choice* of rule (3). Note that the "$\times$" connective is allowed to appear in the head of rules only; it is used to define a preference relation that allows to *select* some of the answer sets of a program by using ranking of literals in the head of the rules, on the basis of a given strategy.

To this end, the semantics of an LPOD program is given in terms of a preference criterion over answer sets. The formal definition of answer sets in LPOD is based on the concept of *split programs* [7]: Given a rule r as in (3), we define for $1 \leq k \leq l$ the *k-th option* of r as the rule

$$r_k \ = \ c_k \leftarrow body(r), not\ c_1, not\ c_2, \ldots, not\ c_{k-1}.$$

Then, P' is some *split program* of an LPOD program P, if it is obtained from P by replacing each rule in P by one of its options. With this concept, Brewka defines in [5] an answer set of an LPOD program P as a consistent answer set of some split program P' of P.

For defining preferred answer sets, Brewka [5] introduces the notion of *degree of satisfaction*: An answer set S satisfies a rule as in (3)

- to degree 1, if $body^+(r) \not\subseteq S$ or $body^-(r) \cap S \neq \emptyset$, and otherwise,
- to degree $d = \min\{k \mid c_k \in S\}$.

The degree of rule r in answer set S is denoted by $deg_S(r)$. Intuitively, the degrees can thus be considered as penalties: the higher the degree, the less we are satisfied about the choice. Brewka shows in [5] that every answer set satisfies all program rules to some degree.

Degrees can be used in various ways for defining a preference relation over answer sets. As an example, we give the definition for the well-known *Pareto* criterion: An answer set S_1 of an LPOD program P is *Pareto-preferred* to another one S_2 ($S_1 >_p S_2$) if there is a rule $r \in P$ such that $deg_{S_1}(r) < deg_{S_2}(r)$ and for no $r' \in P$ we have $deg_{S_1}(r') > deg_{S_2}(r')$. Then, an answer set S of P is *Pareto-preferred* among all answer sets, if there is no answer set S' of P that is Pareto-preferred to S.

For extending the expressive power of LPOD programs in view of our application, we take advantage of the concept of *cardinality constraint* [8,9]. Syntactically, a cardinality constraint is a complex literal of the form:

$$l\ \{a_1, \ldots, a_m\}\ u \tag{4}$$

where l and u are two integers giving a *lower* and *upper* bound, respectively, on the number of satisfied literals within the constraint[2]. For a cardinality constraint C as in (4), we let $lit(C)$ denote its set of literals $\{a_1, \ldots, a_m\}$ and let $lb(C) = l$ and $ub(C) = u$. C is *satisfied* by a set of literals S, if

$$lb(C) \leq |lit(C) \cap S| \leq ub(C) \ .$$

Whenever bound l or u is missing, it is taken to be 0 or $|lit(C)|$, respectively. In what follows, we restrict ourselves to cardinality constraints, C, such that $0 \leq lb(C) \leq ub(C) \leq |lit(C)|$. For defining answer sets of programs with cardinality constraints, we follow the approach taken in [9].

3 From LPOD to S-LPOD

In what follows, we present a straightforward extension of LPOD that allows us to express preferences on sets of atoms.

In policy enforcement contexts [4], it is rather unintuitive that the syntax of rules of the form in (3) requires us to impose a total preference ordering over actions (as with $c_1 \times \ldots \times c_j$), in particular when objects on which actions have to be executed (e.g. devices, users, etc.) are classified on the basis of some given parameters. In similar cases, such total ordering may be unrealistic or even unacceptable.

We thus need to introduce a syntactic variation to the rules of (3) in order to accommodate partial preference orderings among actions, according to the classification of objects involved.

Definition 1. *An S-LPOD program consists of S-LPOD rules of the form*

$$C_1 \times \ldots \times C_l \leftarrow A_1, \ldots, A_m, not\ B_1, \ldots, not\ B_n \tag{5}$$

where each A_i, B_j, C_k is a cardinality constraint for $0 \leq i \leq m$, $0 \leq j \leq n$, and $0 \leq k \leq l$.

A single literal l can be represented by the cardinality constraint $1\{l\}$, as we illustrate below.

For a set of literals S and a cardinality constraint C, define the number of literal of C that are in S as $sel(C, S) = |S \cap lit(C)|$. Then, given a set of literals S, the intuitive reading of the rule head of an S-LPOD rule as in (5) can be given as follows:

- if $lb(C_1) \leq sel(C_1, S) \leq ub(C_1)$, then choose $sel(C_1, S)$ elements of $lit(C_1)$, otherwise
- if $lb(C_2) \leq sel(C_2, S) \leq ub(C_2)$, then choose $sel(C_2, S)$ elements of $lit(C_2)$, otherwise
- ...

[2] The interested reader may note that we confine ourselves to positive literals within cardinality constraints. As detailed below, this is motivated by our application.

– if $lb(C_l) \leq sel(C_l, S) \leq ub(C_l)$, then choose $sel(C_l, S)$ elements of $lit(C_l)$,
– otherwise an incoherent situation is obtained.

The number of elements selected from the chosen cardinality constraint is determined by S. It is nonetheless non-deterministic insofar that different choices of S yield different selections.

The definition of an *option* as well as that of a *split program* carry over from LPOD programs to S-LPOD programs. Answer sets of (split) programs with cardinality constraints are defined as in [9]. Let us illustrate this by building split programs of a S-LPOD along those definitions.

Example 1. Let program P consist of the rules:

$$r_1 : \ 1\{a, b\}1 \ \times \ \{c, d, e\}. \qquad r_2 : \ 1\{b, c, d\} \ \times \ 1\{a, f\}.$$

We obtain 4 split programs:

$$\begin{array}{ll} P_1' : 1\{a, b\}1. & P_2' : 1\{a, b\}1. \\ \quad 1\{b, c, d\}. & \quad 1\{a, f\} \leftarrow not\ 1\{b, c, d\}. \\ P_3' : \{c, d, e\} \leftarrow not\ 1\{a, b\}1. & P_4' : \{c, d, e\} \leftarrow not\ 1\{a, b\}1. \\ \quad 1\{b, c, d\}. & \quad 1\{a, f\} \leftarrow not\ 1\{b, c, d\}. \end{array}$$

We obtain the following answer sets for program P[3]:

$$\{a\},\ \{b\},\ \{c\},\ \{d\},\ \{f\}, \{a, c\},\ \{a, d\},\ \{a, f\}, \{b, c\}, \{b, d\},$$
$$\{c, e\}, \{c, d\},\ \{d, e\},\ \{e, f\}, \{a, c, d\},\ \{b, c, d\},\ \{c, d, e\}$$

Hence, as with standard LPOD programs, an answer set of an S-LPOD program is simply an answer set of one of its split programs.

To complete the semantics of S-LPOD programs, we first have to account for the definition of the *degree of satisfaction*:

Definition 2. *A set of literals S satisfies a rule as in (5)*

– *to degree 1, if A_i is not satisfied by S for some $0 \leq i \leq m$ or B_j is satisfied by S for some $0 \leq j \leq n$, and otherwise,*
– *to degree $d = \min\{k \mid lb(C_k) \leq sel(C_k, S) \leq ub(C_k)\}$.*

As above, we denote the degree of rule r in answer set S as $deg_S(r)$. As with standard LPOD, our extended definition assures that if an answer set S of an S-LPOD program exists, then S satisfies all rules of P to some degree.

As well, we can use the degree of satisfaction to induce different preference criteria on the answer sets of an S-LPOD program. In particular, the criterion of *Pareto-preference* given above carries over from LPOD to S-LPOD.

Example 2. Consider again the program P given in Example 1. All Pareto-preferred answer sets satisfy both rules of program P with degree 1:

$$\{a, c\},\ \{a, d\},\ \{b\},\ \{b, c\},\ \{b, d\},\ \{a, c, d\},\ \{b, c, d\}.$$

[3] Each of the answer set reported is an answer set of at least one of the split programs. This is a necessary condition to be answer set of the original program [5].

Thus, we can have more than one preferred answer set and each of them is also an answer set of some split program of the original program.

Finally, let us show that S-LPOD is *conservative* insofar as it corresponds to LPOD whenever we have no cardinality constraints. To see this, consider rule (5), where each cardinality constraint is of the form $1\{l\}$ for some literal $l \in \mathcal{L}$:

$$1\{c_1\} \times \ldots \times 1\{c_l\} \leftarrow 1\{a_1\}, \ldots, 1\{a_m\}, \mathit{not}\ 1\{b_1\}, \ldots, \mathit{not}\ 1\{b_n\}. \quad (6)$$

A set of literals S satisfies such a rule r

- to degree 1, if $1\{a_i\}$ is not satisfied by S for some $0 \leq i \leq m$ or $1\{b_j\}$ is satisfied by S for some $0 \leq j \leq n$, and otherwise,
- to degree $d = \min\{k \mid lb(c_k) \leq sel(c_k, S) \leq ub(c_k)\}$.

In this special case, we have $sel(l, S) = |S \cap lit(l)| = |S \cap \{l\}|$, and $lb(l) = 1$ and $ub(l) = |lit(l)|$. While the first condition is equivalent to $body^+(r) \not\subseteq S$ or $body^-(r) \cap S \neq \emptyset$, the latter gives $d = \min\{k \mid 1 \leq |S \cap \{l\}| \leq |lit(l)|\}$. In order to respect the bounds $1 \leq |S \cap \{l\}| \leq 1$, we must have $l \in S$, so that the degree of satisfaction of rule r is $d = \min\{k \mid l \in S\}$, which is what we have in the definition of degree of satisfaction for LPODs.

4 From S-LPOD to SR-LPOD

We have seen in Example 1 that S-LPOD programs may yield many answer sets, among which one may still find a substantial number of preferred answer sets. This is even more severe in practice. In fact, in practice, it is also very natural to impose additional preferences among S-LPOD rules.

As before, it turns out that ASP-techniques can be composed in a quite straightforward way in order to obtain an extension encompassing the desired features. To this end, we take advantage of *ordered logic program*, being a pair $(P, <)$, where P is a logic program and $< \subseteq P \times P$ is a strict partial order. Given, $r_1, r_2 \in P$, the relation $r_1 < r_2$ expresses that r_2 has *higher priority* than r_1. Then, an *SR-LPOD program* is an ordered logic program $(P, <)$, where P is an S-LPOD program. As before, the formation of preferred answer sets can be made precise in different ways. Among them, we follow the proposal in [6] by using the extended definition of the Pareto-preference criteria proposed in [6, Definition 9]: An answer set S_1 of an LPOD program P is *Pareto-preferred* to another one S_2 wrt program P, written as $S_1 >_{pr} S_2$, if

1. there is a rule $r \in P$ such that $deg_{S_1}(r) < deg_{S_2}(r)$ and
2. for each $r' \in P$ such that $deg_{S_1}(r') > deg_{S_2}(r')$, there is some rule r'' such that $r' < r''$ and $deg_{S_1}(r'') < deg_{S_2}(r'')$.

We found out that this definition is applicable to SR-LPOD[4] and it allows us to obtain a more fine-grained ordering on answer sets as with S-LPOD program,

[4] The interested reader may note that we only consider *static* preferences among S-LPOD rules, i.e. meta-preference statements of the form $r_1 < r_2$ with empty body.

in that it may introduce additional preferences among answer sets that were considered incomparable or equally preferred according to the original definition of Pareto-preference criteria given in [5], even when preferences on sets of objects are expressed.

We also show that the ordering relation on answer sets of an (S-)LPOD program P is preserved if we add to P preferences on its (S-)LPOD rules. In fact, the following proposition holds:

Proposition 1. *Let S_1 and S_2 be answer sets of an (S-)LPOD program P. Then $S_1 >_p S_2$ implies $S_1 >_{rp} S_2$*

Proof. Let us suppose that $S_1 >_{rp} S_2$ does not hold, and show that $S_1 >_p S_2$ does not hold too. $S_1 >_{rp} S_2$ does not hold if one of the properties in its definition do not hold, i.e.

1. $\forall r \in P,\ deg_{S_1}(r) \geq deg_{S_2}(r)$
2. $\exists r' \in P :\ deg_{S_1}(r') > deg_{S_2}(r')$ and $\forall r'' > r',\ deg_{S_1}(r'') \geq deg_{S_2}(r'')$.

In the first case, we can immediately conclude that $S_1 >_p S_2$ does not hold by the first part of the definition of preference relation $>_p$.

In the second case, we have that whenever such r' exists, $deg_{S_1}(r') > deg_{S_2}(r')$ holds, and thus $S_1 >_p S_2$ by the second part of the definition of preference relation $>_p$.

Example 3. Let us consider the S-LPOD program P in Example 1. The ordering relation among the answer sets of P can be represented by considering the following three sets:

$$AS_1 = \{\{a,c\},\ \{a,d\},\ \{b\},\ \{b,c\},\ \{b,d\},\ \{a,c,d\},\ \{b,c,d\}\}$$
$$AS_2 = \{\{a\},\ \{c\},\ \{d\},\ \{a,f\},\ \{c,e\},\ \{c,d\},\ \{d,e\},\ \{c,d,e\}\}$$
$$AS_3 = \{\{f\},\ \{e,f\}\}$$

According to the ordering relation derived from the original definition of Pareto-preference criteria in Section 2, we have $S_i > S_j > S_k$ for $S_i \in AS_1, S_j \in AS_2,\ S_k \in AS_3$. Two answer sets in the same partition are considered incomparable or equally preferred.

If we add the meta-preference on S-LPOD rules of P expressed by $r_1 < r_2$, a more fine-grained ordering is achieved and we can identify one partition more, thus specializing the preference relation among previously incomparable answer sets, as follows:

$$AS_1 = \{\{a,c\},\ \{a,d\},\ \{b\},\ \{b,c\},\ \{b,d\},\ \{a,c,d\},\ \{b,c,d\}\}$$
$$AS_2 = \{\{c\},\ \{d\},\ \{c,e\},\ \{c,d\},\ \{d,e\},\ \{c,d,e\}\}$$
$$AS_3 = \{\{a\},\ \{a,f\}\} \qquad AS_4 = \{\{f\},\ \{e,f\}\}$$

where $S_i > S_j > S_k > S_l$ for $S_i \in AS_1,\ S_j \in AS_2,\ S_k \in AS_3,\ S_l \in AS_4$.

One may argue that these new meta-preferences among S-LPOD rules do not significantly change the solution, since the Pareto-preferred answer sets of P are

the same. But suppose that there are integrity constraints preventing us from considering any of the most preferred answer sets as a solution, e.g. the following three constraints are added to program P:

$$r_{c1}: \quad \leftarrow a, c. \qquad r_{c2}: \quad \leftarrow b. \qquad r_{c3}: \quad \leftarrow d.$$

As a result, all answer sets in AS_1 are eliminated and some in AS_2 are the preferred ones; preference ordering among them has been refined by the new preference relation among S-LPOD rules, so that the solution is reduced to answer sets $\{c\},\ \{c, e\}$ as the preferred ones.

The following example illustrates how Pareto-preference including preferences among rules can be meaningful even with simple LPOD programs.

Example 4. Let program P_{pref} consist of the LPOD rules:

$$\begin{array}{llll} r_1: & a \times c. & r_{p_1}: r_3 > r_1. & r_{c1}: \leftarrow a, d. \\ r_2: & b \times d. & r_{p_2}: r_4 > r_1. & \\ r_3: & b \times a. & r_{p_3}: r_3 > r_2. & \\ r_4: & d \times c. & r_{p_4}: r_4 > r_2. & \end{array}$$

where rules r_{pi} represent preference relations among rules r_k of P_{pref}.

We can compute 16 split programs[5] obtaining from them the following answer sets for the original program P_{pref}:

$$\begin{array}{ll} S_1 = \{a, b, c\} & deg_{S_1}(r_1) = deg_{S_1}(r_2) = deg_{S_1}(r_3) = 1,\ deg_{S_1}(r_4) = 2 \\ S_2 = \{b, c, d\} & deg_{S_2}(r_2) = deg_{S_2}(r_3) = deg_{S_2}(r_4) = 1,\ deg_{S_2}(r_1) = 2 \\ S_3 = \{b, c\} & deg_{S_3}(r_1) = deg_{S_3}(r_4) = 2,\ deg_{S_3}(r_2) = deg_{S_3}(r_3) = 1 \end{array}$$

According to the Pareto-preference ordering, under LPOD semantics, S_1 and S_2 are the preferred answer sets for P_{pref}; moreover, $S_1 >_p S_3$ and $S_2 >_p S_3$. The extended notion of preference relation on LPOD rules (expressed in rules $r_{p_i},\ i = 1..4$), gives us a more fine-grained ordering on answer sets S_1 and S_2 that were incomparable under the LPOD semantics, in that $S_2 >_{pr} S_1$.

As a consequence, only S_2 results being the Pareto-preferred answer set of P_{pref} according to $>_{pr}$ ordering relation.

5 Application to Policy Enforcement

As illustrated in [4], PPDL is a rather simple, easy-to-grasp language allowing to define policies and consistency mechanisms in a transparent and easy way by keeping the so-called *business logic* outside the specific system representation. PPDL specifications are directly mapped into ASP and can thus be computed very efficiently by invoking performant ASP solvers [6].

Although the encoding of PPDL into LPOD proposed in [4] is intuitive and computationally easy to be automatized, it requires us to impose a total preference ordering over actions to be blocked in a single constraint specification.

[5] Note that only the coherent ones are used for computing solutions.

Such a total ordering can be unrealistic or even unacceptable in applications, as it would force us to specify all possible combinations of totally ordered list of actions. We could need to group objects and consequently actions performed on those objects, according to some common properties, thus adding a level of non-determinism to the choice of which actions to block in order to solve a conflict but keeping the PPDL specification intuitive and the mapping into ASP computationally simple. As an example, consider again a Resource Manager. We may want to tell that a clerk should be prevented from accessing critical resources, but resources availability is to be granted to managers[6]. The above mentioned scenario suggests that we need to introduce a syntactic variation to policy and monitor specification, in order to accommodate partial preference orderings among sets of actions.

It is worth mentioning the fact that, in our policy specification, we allow only positive atoms to appear in the constraints, as each literal represents an action and we do not consider the case in which a set of events causes an action *not to be executed*. Of course this is a possibility and things could be generalized, but we don't deal with this case here.

According to the LPOD extensions we investigated in Section 3 and 4, we now extend the language of PPDL, mentioned in Section 1, into SR-PPDL, by providing a more general definition of a monitor expressing preferences on sets of actions that have to be blocked to solve conflicts arisen from policy enforcement.

Let $\langle A, < \rangle$ be a partially ordered set of actions. We define a level mapping ℓ as follows.

- $\ell(a) = 1$ iff $\nexists a', a' < a$.
- $\ell(a) = i + 1$ iff $\max\{\ell(a') : \ a' < a\} = i$.

The level function partitions A into disjoint sets of actions: $A = A_1 \cup \ldots \cup A_r$, where each A_i contains actions with the same preference level i and $A_g \cap A_l = \emptyset$ for all $g \neq l$.

The preference relation defined by $\langle A, < \rangle$ can be expressed by the extended syntax of SR-PPDL monitor rule (extending the one proposed in [4]) as follows:

$$r \ : \ \textbf{never} \ l_1[A_1]u_1 \times \ldots \times \ l_r[A_r]u_r \ \textbf{if} \ C. \tag{7}$$

where each A_i represents a set of atoms $\{a_1^i, a_2^i, \ldots, a_m^i\}$, C is a Boolean condition and each element $l_i[A_i]u_i$ represents a cardinality constraint of the form in Equation (4).

Given that D_i is the set of actions in A_i triggered by the policy application, the cardinality constraint $C(A_i) = l_i\{a_1^i, a_2^i, \ldots, a_m^i\}u_i$ is satisfied if $l_i \leq |D_i| \leq u_i$. For each constraint $C(A_i)$ that is satisfied, we define the set of actions to be blocked X_i as the minimum subset of D_i for which $|D_i - X_i| \leq l_i - 1$. As a consequence, we have that, for each of these X_i, $|X_i| = |D_i| - l_i + 1$.

Equation (7) tells us that when *all* cardinality constraints $C(A_i)$, $i = 1..r$ are satisfied[7], then actions in D_1, actions in D_2, ..., actions in D_r cannot be

[6] A complete example in this context will be detailed later on in this section.

[7] Otherwise, if at least one of the $C(A_i)$ is not satisfied, there is no conflict and rule r of the form in Equation (7) is not triggered.

executed together *and*, in case of constraint violation, $|X_1|$ actions in D_1 should be preferably blocked; if it is not possible, block $|X_2|$ actions in D_2; ...; if all of the actions in $D_j,\ j = 1..r-1$ must be performed, block $|X_r|$ actions in D_r.

In this way, the total ordering among conflicting actions to be blocked can be released by admitting that actions at a certain level i in the head of an SR-LPOD rule can be non-deterministically chosen from a set $D_i \subseteq A_i$ of equally preferred actions triggered by the policy, given that $C(A_i)$ is satisfied and all other actions in D_j with $l_j \leq |D_j| \leq u_j$ and level $j < i$, must be executed.

To express such non-determinism, we translate the SR-PPDL rule in Equation (7) into SR-LPOD by using cardinality constraints for each set of equally preferred literals. Thus, according to the original PPDL encoding [4], given that

$$A_1 = \{a_1^1, a_2^1, \ldots, a_g^1\} \qquad A_2 = \{a_1^2, a_2^2, \ldots, a_h^2\} \qquad \ldots \qquad A_r = \{a_1^r, a_2^r, \ldots, a_m^r\}$$

each rule of the form in Equation (7) will result into $\prod_{i=1..r}(u_i - l_i + 1)$ SR-LPOD rules representing all possible combination of sets of elements in the head of the SR-LPOD rules:

$$
\begin{array}{l}
l_1\{block(a_1^1),\ block(a_2^1),\ \ldots,\ block(a_g^1)\}l_1 \times \\
l_2\{block(a_1^2),\ block(a_2^2),\ \ldots,\ block(a_h^2)\}l_2 \times \\
\ldots \times \\
l_r\{block(a_1^r),\ block(a_2^r),\ \ldots,\ block(a_m^r)\}l_r \ \leftarrow \ l_1\{exec(a_1^1), \ldots,\ exec(a_g^1)\}l_1, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ l_2\{exec(a_1^2), \ldots,\ exec(a_h^2)\}l_2, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ \ldots \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ l_r\{exec(a_1^r), \ldots,\ exec(a_m^r)\}l_r,\ C. \\
\ldots \\
u_1\{block(a_1^1),\ block(a_2^1),\ \ldots,\ block(a_g^1)\}u_1 \times \\
u_2\{block(a_1^2),\ block(a_2^2),\ \ldots,\ block(a_h^2)\}u_1 \times \\
\ldots \times \\
u_r\{block(a_1^r),\ block(a_2^r),\ \ldots,\ block(a_m^r)\}u_r \ \leftarrow u_1\{exec(a_1^1), \ldots,\ exec(a_g^1)\}u_1, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ u_2\{exec(a_1^2), \ldots,\ exec(a_h^2)\}u_2, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ \ldots \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ u_r\{exec(a_1^r), \ldots,\ exec(a_m^r)\}u_r,\ C. \\
accept(A) \ \leftarrow \ exec(A),\ not\ block(A). \\
\leftarrow block(A),\ not\ exec(A).
\end{array}
$$

The last constraint has been introduced into the mapping from SR-PPDL to SR-LPOD in order to assure that, in non-determinism induced by cardinality constraints on sets, actions blocked are among those triggered by the policy.

Corresponding split programs are built in the same way as illustrated in Section 2. This may generate a lot of possibilities, further reduced when we introduce a preferential ordering of the form $r_i > r_j$ where r_i and r_j are SR-PPDL rules of the form in Equation (7). Combination of our LPOD extensions into the high-level policy language is illustrated in the following example.

Example 5. Let us consider the problem of allocation of resources a, b, c and d among two users, $u1$ and $u2$. Resources a and b are critical (actions corresponding

to the assignment of a and b should be preferentially blocked in case conflicts arise), user $u2$ is to be preferentially served over $u1$.

The corresponding monitor rules look like:

$$r_1 \ : \textbf{never}\ 1[ass(u1,R)]1 \times 1[ass(u2,R)]1.$$
$$r_2 \ : \textbf{never}\ 1[ass(U,a),ass(U,b)]2 \times 1[ass(U,c),ass(U,d)]2.$$

where U and R are grounded on the set of users and resources, respectively.

Suppose that the policy application yields user $u1$ to obtain resources b, c, d, and user $u2$ to obtain resources a, b, c[8] and $u1$ needs at least one resource among b and c.

The resulting SR-LPOD program $P_{sr-lpod}$ is as follows:

$$r_1^1 : 1\{block(u1,b)\}1 \times 1\{block(u2,b)\}1.$$
$$r_1^2 : 1\{block(u1,c)\}1 \times 1\{block(u2,c)\}1.$$
$$r_2^1 : 1\{block(u1,a), block(u1,b)\}1 \times 2\{block(u1,c), block(u1,d)\}2.$$
$$r_2^2 : 2\{block(u2,a), block(u2,b)\}2 \times 1\{block(u2,c), block(u2,d)\}1.$$
$$\leftarrow block(u1,b), block(u1,c).$$
$$\leftarrow block(U,R),\ not\ exec(U,R),\ res(R),\ usr(U).$$
$$accept(U,R) \leftarrow not\ block(U,R),\ res(R),\ usr(U).$$

We obtain three answer sets of $P_{sr-lpod}$:

$$S_1 = \{block(u1,b), block(u2,c)\}$$
$$S_2 = \{block(u1,c), block(u1,d), block(u2,b), block(u2,c)\}$$
$$S_3 = \{block(u1,b), block(u2,a), block(u2,b), block(u2,c)\}$$

with $S_3 >_p S_1$. Thus, S_2 and S_3 are the preferred answer sets for $P_{sr-lpod}$ in terms of blocked assignments.

Suppose we now add a rule preference $r_2 > r_1$ on rules of the monitor. This means that, in their grounded instances, each of the rules r_2^i is preferred to each of the rules r_1^j. According to relation $>_{rp}$, we now have $S_3 >_{rp} S_1$[9] and $S_3 >_{rp} S_2$, thus obtaining only S_3 as the preferred answer set.

To accomplish specific systems requirements, additional constraints could be added, such as that each user has to be assigned to at least one resource, or that a resource cannot be assigned to two different users, thus restricting the set of admissible solutions.

It's easy to imagine that when a wide number of combinations are possible according to how resources/users are grouped into sets, introducing a further level of preferences on rules that determine (S-)LPOD preferences, can results in more accurate solutions by reducing the set of Pareto-preferred solutions.

[8] For simplicity, we focus on the consistency monitor specification omitting policy rules. This does not change the way priorities are computed, cause not triggered rules have degree equal to 1 by definition.

[9] Note that the Pareto-preference relation $>_p$ is preserved.

6 Conclusion

We have considered advanced policy description specifications based on advanced semantics of Answer Set Programming. Our analysis is aimed at providing a tool to enforce complex policy consistency mechanisms, enriched with qualitative preferential information by using the high-level policy description language PPDL investigated in [4]. To this end, we extended the logical formalism by allowing ordered disjunctions over cardinality constraints and we used a rule-based Pareto-preference criterion for distinguishing preferred answer sets. Next step is to extend the PPDL language syntax in this direction, by mapping extended monitor constructs of the resulting SR-PPDL into SR-LPOD.

We believe in the potential of high-level specification languages to control and monitor complex systems efficiently. In fact, the proposed extension to qualitative preference handling from policy and monitor enforcement perspectives could enable us to find new contexts of application in other fields of AI.

Future work will address implementation issues: we want to adapt the compilation technique proposed in [6] to our approach. Also, it is worthwhile to check under which restrictions a specification can be compiled in a normal logic program (without any need for genuine disjunctions).

References

1. Chomicki, J., Lobo, J., Naqvi, S.: Conflict resolution using logic programming. IEEE Transactions on Knowledge and Data Engineering 15(1), 244–249 (2003)
2. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press, Cambridge (2003)
3. Mileo, A.: Preference Specification and Enforcement in Declarative Policies. PhD thesis, Universitá degli Studi di Milano (2006)
4. Bertino, E., Mileo, A., Provetti, A.: PDL with preferences. In: Proc. of POLICY 2005 (2005)
5. Brewka, G.: Logic programming with ordered disjunction. In: Proc. of AAAI02. Extended version presented at NMR02 (2002)
6. Brewka, G., Niemelä, I., Syrjänen, T.: Implementing ordered disjunction using answer set solvers for normal programs. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, pp. 444–455. Springer, Heidelberg (2002)
7. Sakama, C., Inoue, K.: An alternative approach to the semantics of disjunctive logic programs and deductive databases. J. Autom. Reasoning 13(1), 145–172 (1994)
8. Simons, P., Niemelä, I., Soininen, T.: Extending and implementing the stable model semantics 138(1-2), 181–234 (2002)
9. Liu, L., Truszczynski, M.: Properties of programs with monotone and convex constraints. In: Proc. of AAAI05, pp. 701–706 (2005)

A Qualitative Hidden Markov Model for Spatio-temporal Reasoning

Zina M. Ibrahim, Ahmed Y. Tawfik, and Alioune Ngom

University of Windsor, 401 Sunset Avenue,
Windsor, Ontario N9B 3P4, Canada
{ibrahim,atawfik,angom}@uwindsor.ca

Abstract. We present a Hidden Markov Model that uses qualitative order of magnitude probabilities for its states and transitions. We use the resulting model to construct a formalization of qualitative spatio-temporal events as random processes and utilize it to build high-level natural language description of change. We use the resulting model to show an example of foreseen usage of well-known prediction and recognition techniques used in Hidden Markov Models to perform useful queries with the representation.

1 Introduction and Motivation

Hidden Markov Models (HMMs) have been successful in representing random processes and acquiring useful characteristic in a mathematically tractable way [14]. The model studies such processes by obtaining as input a discrete time sequence representing the observable output emitted by the process over time [16]. Using the model, it is possible to perform prediction and recognition tasks. The stochastic model has been used in various applications such as face recognition [2], speech recognition [18], spatio-temporal pattern recognition in image processing [15] and Bioinformatics [12].

The mechanisms offered by HMMs are readily available if one can have (or learn) the values associated with the probability distribution required to construct the model. Although this is possible for many applications, this luxury may not be available in many other applications and a complete specification of the probability values of the events constituting the model is not achievable.

In this paper, we propose a qualitative Hidden Markov Model which uses order of magnitude probabilities [7] instead of numerical probabilities to overcome the difficulty of using HMMs in applications where numerical values of the probability distributions are not obtainable.

One such domain of applications is that of qualitative spatio-temporal representation and reasoning, which has flourished because the epistemic nature of spatio-temporal information usually renders it vague and highly-dynamic, making the numerical values of the attributes in concern mostly unknown. This is why numerical approaches to reasoning about the spatio-temporal domain are limited, and qualitative methods have prevailed [1].

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 707–718, 2007.

The general skeleton of a qualitative representation consists of a model of real-world objects as *abstract entities*, which take the form of points [5] or regions of space [17]. The formalism then chooses one or more spatial features of interest to be represented (e.g. topology [17] [5], orientation [10] or distance [13]) and constructs a set of *relations* capturing all the possible interactions between two abstract entities with respect to the chosen spatial feature [4].

Reasoning is carried out via queries that perform prediction and recognition tasks, by studying the different relations that may hold among the objects at different times [4].

Because uncertainty prevails in the spatio-temporal domain, incorporating reasoning techniques that are capable of dealing with uncertain information has been a focus in the qualitative spatio-temporal community. For example [9] incorporates fuzzy-sets in the formulation of the queries about the set of spatial relations in order to accommodate vagueness, which although yields a robust way to deal with uncertainty, has not been expanded to tackle predictive queries, and is mainly concerned with recognitive queries. Also, [3] feeds the transitions between spatial relations as evidences to a Bayesian network, which in turn provides the probabilities of future transitions. This work however, is limited to applications where it is possible to learn the numerical probabilities of the transitions through some form of sensors, e.g. robot navigation [3].

In this paper, we apply the Qualitative Hidden Markov Model constructed to the spatio-temporal domain and consequently use it to reason about motion by constructing a qualitative HMM for a topology-based qualitative spatio-temporal representation.

The paper is structured as follows. We begin in section 2 by an overview of the spatio-temporal calculus on which our Qualitative HMM will be applied. Section 3 reviews the concepts of order of magnitude probabilities which will be used as the building blocks to our Qualitative HMM. In section 4, we present our Qualitative HMM, HMM_ϵ, and equip it with a qualitative algorithm to perform recognition tasks. In section 5, we show how HMM_ϵ can be used to model the evolution of qualitative spatio-temporal relations between objects as they move. In section 5.2, we show an example of possible reasoning techniques the model is capable of carrying out. We conclude in section 6 by detailing our future work.

2 The RCC8 Calculus

A topology-based spatial theory which abstracts all physical entities to regions of space whose exact size, shape and location are irrelevant, and uses the notion of two regions being connected to construct a set of *jointly-exhaustive and pairwise disjoint* (JEPD) [1] [4] qualitative spatial relations to hold between any two regions [17]. For two regions x and y, the diadic relation $C(x,y)$ (x is connected to y) holds if regions x and y share some common parts. From this notion, eight

[1] Meaning that together, the relations in the set represent all the possible interactions with respect to the chosen spatial feature, and that no two relations in the set can hold at the same time for the same spatial objects.

topological relations have been defined. They are: DC(x,y) (x is discoonected from y), EC(x,y) (x is externally connected to y), PO(x,y) (x partially overlaps y), EQ(x,y) (x is equal to y), TPP(x,y) (x is a tangental proper part of y), its inverse TPPI(x,y) (y is a tangental proper part of x), NTPP(x,y) (x is a non-tangental proper part of y), and NTPPI(x,y) (y is a non-tangental proper part of x). The complete RCC8 set is shown in figure 1. In this work, we use the notation $r_{[t,t+\Delta]}(x,y)$ to denote that RCC8 relation r holds between region x and y during the interval $[t,t+\Delta]$.

RCC8 possesses continuity properties captured by its *conceptual neighborhood graph* (CNG) [8] shown in figure 2. The graph captures the notion of somewhat similar relations possessing somewhat similar behavior. It can be seen from the figure that any two relations are neighbors in a CNG if they can be directly transformed into one another by a continuous deformation (i.e. decrease or increase of size or distance) of the objects. For example, EC and PO are conceptual neighbors; there exists a direct transformation that can change the relation between two regions x and y from EC to PO and vice versa. On the other hand, DC and PO are not conceptual neighbors, because the relation between two regions x

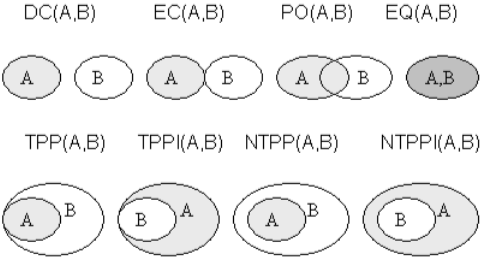


Fig. 1. The RCC8 Set

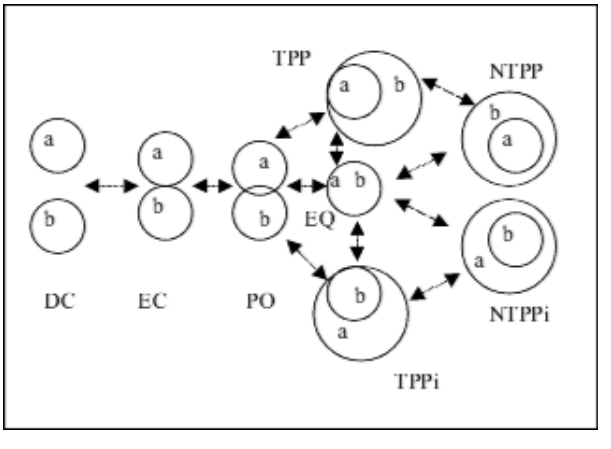


Fig. 2. The RCC8 Set Conceptual Neighborhood Graph

and y to change from DC to PO or vice versa, the regions have to go through a transformation which forces them to be externally connected (EC holds between x and y) at some point in order for the transformation to take place.

3 A Theory of Qualitative Probabilities

[7] introduced a qualitative theory of probability to model causality and belief. In this work, the order of magnitude of probability is used to represent degrees of belief, rather than numerical values that describe that frequency of occurrence of an event.

The formalism represents the probability of a proposition ω, $\mathrm{P}(\omega)$, by a polynomial function of one unknown, ϵ, an infinitesimally small positive number ($0 < \epsilon < 1$). The rank κ of a proposition ω represents the degree of incremental surprise or abnormality associated with finding ω to be true [7]. It is represented by the power of the most significant ϵ-term in the polynomial representing $\mathrm{P}(\omega)$ (the lowest power of ϵ in the polynomial).

The idea behind infinitesimal representations is that $\mathrm{P}(\omega)$ is of the same order as ϵ^n where $\kappa(\omega) = n$. In other words:

$$\epsilon < \frac{P(\omega)}{\epsilon^n} \leq 1$$

Where ϵ^n is the most significant ϵ-term of the polynomial representing $\mathrm{P}(\omega)$. The ranking function $\kappa(\omega)$ is defined below:

$$\kappa(\omega) = \begin{cases} min\{n : \lim_{\epsilon \to \infty} P(\omega)/\epsilon^n \neq 0\}, & \text{if } \mathrm{P}(\omega)>0 \\ \infty, & \text{otherwise.} \end{cases}$$

ϵ-semantics is useful because it provides an abstraction which only requires specifying the κ values of the propositions, which is an easier task than specifying the exact probabilities associated with the occurrence of the proposition. The κ values are in turn representative of the interval into which the probability falls.

Properties of the order of magnitude probabilities are directly extracted from those of probability theory. There exists a mapping between the two which is done by replacing the addition operators of probability theory by the *min* operator, and multiplication by addition. We give below the properties that are relevant to this work, along with their probability theory equivalents. For details on how these properties are obtained, the reader may refer to [7].

$$\kappa(\varphi) = min_{\omega \models \varphi} : \quad P(\varphi) = \sum_{\omega \models \varphi} P(\omega) \tag{1}$$

$$\kappa(\varphi) = 0 \vee \kappa(\neg\varphi) = 0 : \; P(\varphi) + P(\neg\varphi) = 1 \tag{2}$$

$$\kappa(\psi|\varphi) = \kappa(\psi \wedge \varphi) - \kappa(\varphi) : \; P(\psi|\varphi) = P(\psi \wedge \varphi)/P(\varphi) \tag{3}$$

4 A Qualitative Hidden Markov Model

In what follows, we define the building blocks to constructing a Hidden Markov Model which relies on the κ-values of propositions presented in section 3 instead of numerical probabilities. The resulting model is a Qualitative Hidden Markov Model, HMM_ϵ.

As in a classical HMM model, $\text{HMM}_\epsilon = (A,\ B,\ \pi)$ requires five elements for its specification. They are defined below.

1. **The States (Q)**
   The model defines N number of states, i.e. $\text{Q} = q_1, q_2, ..., q_N$, deriving their labels from the set of state alphabet $\text{S} = s_1, s_2, ..., s_N$. The state q at time t is denoted by: $q_t = s_i$, where $1 \leq i \leq N$.
2. **The Alphabet (O)**
   Every state $q_i \in \text{Q}$ emits an output o_i which is the only observable of the model. This results in a sequence of emissions of the model ($\text{O} = o_1, o_2, ..., o_N$). The set of of all possible (and distinct) alphabet symbols $V = v_1, v_2, ..., v_M$ is the domain from which every output $o_i \in \text{O}$ emitted by state $q_i \in \text{Q}$ (where $1 \leq i \leq N$) takes its value.
3. **State Transitions Matrix (A)**
   A is an array storing the degrees of surprise associated with state j following state i.

$$\text{A} = [a_{ij} | a_{ij} = \kappa(q_t = s_j | q_{t-1} = s_i)]$$

4. **Observations Vector (B)**
   B is an array storing the degrees of surprise associated with observing output j being produced from state i, independent of time.

$$\text{B} = [b_i(j) | b_i(j) = \kappa(o_t = v_j | q_t = s_i)]$$

5. **Initial Vector (π)**
   π is an array storing the initial degrees of surprise associated with the states of the model.

$$\pi = [\pi_i | \pi_i = \kappa(q_1 = s_i)]$$

4.1 Assumptions

HMM_ϵ adheres to the assumptions that a classical HMM adheres to as made clear by the discussions below.

Markov Assumption. The Markov assumption states that the next state is only dependent on the current state, with all the previous states being irrelevant. The Markov Assumption for HMM_ϵ is expressed as given below:

$$\kappa(q_t | q_{t-1}, q_{t-2}, ..., q_1) = \kappa(q_t | q_{t-1})$$

Independence Assumption. States that the output observation at time t is only dependent on the current state, with all the previous states, along with their outputs being irrelevant to it.

$$\kappa(o_t|o_{t-1}, o_{t-2}, ..., o_1, q_t, q_{t-1}, ..., q_1) = \kappa(o_t|q_t)$$

Markov Chain Assumptions. HMM_ϵ also adheres to the Markov chain assumptions, which are formalized by reformulating the Markov chain assumptions of classical HMMs via equation (1) given in section 3. The assumptions, along with their HMM_ϵ formulations, are given below.

1. The sum of the emission degrees of surprise (κ values) for each state is equal to 1. This is preserved by the property:
   $\kappa(V|q_j) = \min_{v_i \models V} \kappa(v_i|q_j) = 1$
2. The sum of all the transition degrees of surprise (κ values) is 1, which is preserved by the property:
   $\kappa(Q|Q) = \min_{q_i \models Q}(\min_{q_j \models Q} \kappa(q_j|q_i)) = 1$

4.2 Reasoning: A Qualitative Decoding Algorithm

The power of classical HMMs stems from the algorithms associated with the three problems HMMs solve, the evaluation, decoding and learning problems [14]. In this paper, we restrict our discussion to formulating a qualitative equivalent to the algorithm associated with the decoding problem as the other problems are part of our current research.

The Decoding Problem. Given the observation sequence $o = o_1, o_2, ..., o_t$ of length t and a model λ=(A,B,π), the decoding problems is concerned with finding the sequence of states $q = q_1, q_2, ..., q_t$ that was most likely to have produced the observation sequence o (i.e. minimizes the degree of surprise that q was used to generate o).

The Score Function. In order to evaluate candidate sequences of states, we require a quantity representing the degree of surprise associated with the most-likely sequence being one which ends with state $q_t = i$. We denote this quantity by $\delta_t(i)$.

$$\delta_t(i) = min_{q_1 q_2 ..., q_t} \kappa(q_1 ... q_{t-1}, o_1 ... o_t, q_t = i) \tag{4}$$

In order to use the score function to find the best sequence q, we should be be able to answer the question: what is the degree of surprise associated with the most-likely sequence being one which ends with state q_{t+1} being state j, given that the degree of surprise associated with the most-likely sequence being one which ends with state $q_t = i$ is $\delta_t(i)$? The answer is found by induction on the length of the sequence q as shown below.

$\sigma_{(t+1)}(j) = min_{\pi_1, ..., \pi_t} \kappa(o_1 ... o_t, q_1, ..., q_t, o_{t+1}, q_{t+1} = j)$

Substituting in Equation (3) of section 3:

$= min_{q_1,...,q_t}\ [\kappa(o_{t+1},q_{t+1} = j \mid o_1...o_t,\ q_1, ..., q_t) + \kappa(o_1...o_t,q_1, ..., q_t)]$
Taking into account Markov and Independence assumptions
and redistributing the rest:
$= min_{q_1,...,q_t}\ [\kappa(o_{t+1},q_{t+1} = j \mid q_t) + \kappa(o_1...o_{t-1},q_1, ..., q_{t-1},\ o_t,\ q_t)]$
However, the sequence that minimized the degree of surprise was the one that ended with state i and which was given by the equation 4. This makes the above:
$= min_t[\ \kappa(o_{t+1},q_{t+1} = j|q_t = i) + min_{q_1q_2...,q_t}\kappa(q_1...q_{t-1}, o_1...o_t, q_t = i)$
$= min_t[\ \kappa(o_{t+1},q_{t+1} = j|q_t = i) + \sigma_t(i)]$
$= b_j(o_{t+1}) + min_t\ [a_{ij} + \sigma_t(i)]$

$$\sigma_{t+1}(j) = b_j(o_{t+1}) + min_t[a_{ij} + \sigma_t(i)] \tag{5}$$

A Qualitative Viterbi Algorithm. The algorithm keeps track of the argument which has minimized 5 at every time t and state j. For this, a vector $\varrho_t(j)$ is used. Hence, the qualitative viterbi algorithm can be described via the following steps:

1. Initialization

$$\sigma_t(i) = \pi_i + b_i(o_1), 1 \leq i \leq N \tag{6}$$
$$\varrho_1(i) = 0 \tag{7}$$

2. Recursion

$$\sigma_t(j) = b_j(o_t) + min_{1\leq i\leq N}[a_{ij} + \sigma_{t-1}(i)]\ 2 \leq t \leq T, 1 \leq j \leq N \tag{8}$$
$$\varrho_t(j) = \quad argmin_{1\leq i\leq N}[a_{ij} + \sigma_{t-1}(i)] \quad 2 \leq t \leq T, 1 \leq j \leq N \tag{9}$$

3. Termination

$$P^* = \quad min_{1\leq i\leq N}[\sigma_T(i)] \tag{10}$$
$$q_T^* = argmin_{1\leq i\leq N}[\sigma_T(i)] \tag{11}$$

4. Path (state sequence) Backtracking

$$q_t^* = \varrho_{t+1}(q_{t+1}^*)\ t = T-1, T-2, ..., 1 \tag{12}$$

5 HMM$_\epsilon$ for a Qualitative Spatio-temporal Calculus

Given two objects, with one moving with respect to the other (e.g., a car c in a highway h), topology can capture the possible spatial interactions between the two objects using the RCC8 relations, where at any time, some relation r∈RCC8 must hold between c and h.

Because the RCC8 set adheres to the continuity constraints specified by its conceptual neighborhood graph [8], motion will follow a specific set of patterns

which abide by these constraints, and which can be identified in the absence of uncertainty. In other words, motion will have to follow the transitions dictated by the conceptual neighborhood graph of the RCC8 relations. Using this, it is possible to construct natural language verbs describing the motion of the car with respect to the highway during any interval $[t, t+\Delta]$. For instance, the car can *leave* the highway during $[t, t+\Delta]$, moving from NTPP(c,h) at t to DC(c,h) at $t+\Delta$.

We would like to use HMM_ϵ to recognize high-level natural-language verbs of change (e.g. *leave* in the above example) which take place between two regions as they move, from the knowledge of a time-series of snapshots of the qualitative spatial relations (RCC8 relations) that hold between the two regions at different times. This will make possible recognizing the patterns which motion follows in the presence of uncertainty by representing motion as a stochastic process. We call the resulting model $\text{HMM}_{st\epsilon}$, whose constituents are given below.

5.1 $\text{HMM}_{st\epsilon}$

1. **The Alphabet:** Or the domain of the possible outputs, consists of RCC8 set = {DC, EC, PO, EQ, TPP, TPPI, NTPP, NTPPI} given in section 2, which represents the possible spatial relations among two objects.
2. **The States:** The set of states Q is the set of motion verbs, which linguistically describe the patterns that motion forms as two regions move with respect to each other during a given interval, changing the topological relations that hold between them accordingly.

   Using the RCC8 relations, [11] formulates a set of patterns describing the motion of a moving object x with respect to another moving object y during some interval $[t, t + \Delta]$. We will use this set, MC, as the set of states for $\text{HMM}_{st\epsilon}$. It consists of the following patterns:

   Leave x y (read as: x leaves y), *Reach x y* (x reaches y), *Hit x y* (x hits y), *Bounce x y* (x bounces off of y), *Peripheral x y* (x moves alongside the edge of y), *Internal x y* (x moves inside y), *External x y* (x moves outside y), *Expand x y* (x increases in size to overpass y) and *Shrink x y* (x decreases in size to be contained in y).

   Every element $mc_i \in$ MC is described by the predicates $starts(mc_i, [t, t+\Delta], x, y)$ and $ends(mc_i, [t, t+\Delta], x, y)$, where $starts(mc_i, [t, t+\Delta], x, y)$ returns the spatial relations that can hold between spatio-temporal objects x and y at the beginning of the interval $[t, t+\Delta]$ during which the verb mc_i correctly describes the change taking place, while $ends(mc_i, [t, t+\Delta], x, y)$ gives the spatial relations that hold between spatio-temporal objects x and y at the end of the interval $[t, t+\Delta]$[2].

   The motion verbs are given in table 1. In the table, the rows correspond to the RCC8 relation which belongs to the set $starts(mc_i, [t, t+\Delta], x, y)$ while the column corresponds to the RCC8 relation which belongs to the set

[2] The reader may refer to [11] for the formal definitions of the predicates and a more detailed discussion on the construction of the set of patterns from RCC8.

Table 1. The Set MC of Motion Classes

STARTS \ ENDS	DC	EC	PO	TPP	NTPP	EQ	TPPi	NTPPi
DC	EXTERNAL	HIT	REACH					
EC	BOUNCE	PERIPHERAL						
PO	LEAVE		LEAVE OR REACH					
TPP				INTERNAL		EXPAND		
NTPP								
EQ				SHRINK		INTERNAL		
TPPi							INTERNAL	
NTPPi								

$ends(mc_i, [t, t + \Delta], x, y)$. Each intersection of a row and a column presents the resulting motion class mc_i when the corresponding RCC8 relations at the beginning and end of the interval hold. Elements of MC continuity properties similar to those of RCC8, and hence, possess a conceptual neighborhood structure [11].

3. **Qualitative Emissions and Transitions:**
The κ values for the emissions and transitions are obtained by using the natural continuity properties possessed by the sets RCC8 and MC. More specifically, we use the conceptual neighborhood structures of RCC8 and MC to assign weights to spatio-temporal events defined in terms of RCC8 and MC. These weights represent the relative likelihood of the events, which are inversely related to the degree of disbelief function (κ values) we will use in the model to be constructed. In what follows, we take a closer look at this likelihood function and how it relates to [7]'s degrees of disbelief.

Likelihood Function. We define, a *weight* function, ς, to represent the likelihood of an event. ς can take one of the following forms:

(a) The likelihood of an RCC8 relation r being true between two objects undergoing change, at the end of interval $[t, t + \Delta]$, which is the interval where the verb mc_i is known to hold. The likelihood function, in this case, returns the number of ways for which motion class mc_i can end with RCC8 relation r for the interval $[t, t + \Delta]$.

Definition 1. $\varsigma(r,mc_i) = \text{length}[ends(mc_i,[t,t + \Delta], x, y)]$

(b) The likelihood of verb mc_j to hold during interval $[t + \Delta,t + \Delta + \gamma]$, which immediately follows the interval $[t,t + \Delta]$, with interval $[t,t + \Delta]$ being the one during which verb mc_i was known to be true. The likelihood function, in this case, yields the number of ways for which verb mc_j can immediately follow verb mc_i, which is called, according to [11], $(contin\_pairs(mc_i,mc_j))$.

Definition 2. $\varsigma(mc_i,mc_j) = \text{length}(contin\_pairs(mc_i,mc_j))$

Kappa Values. κ, i.e. the degree of disbelief in an event (or a set of events) has the following inverse relation with the weight of the event ς:

Property 1. $\forall mc_i, mc_j, mc_k : \varsigma(mc_i,mc_j) \leq \varsigma(mc_i,mc_k) \rightarrow \kappa(mc_i,mc_j) \geq \kappa\ (mc_i,mc_k)$

Given a set of JEPD relations, RCC8, representing the alphabet of the model, and a set of motion patterns MC, representing the states of the model, algorithm *calculate-*κ allocates *kappa* values to the various transitions of the HMM to be constructed.

The algorithm takes as input two sets, S_1 and S_2, which can either be subsets of RCC8 or of MC, as illustrated below via examples.

The algorithm finds the set w by obtaining the elements with similar weights (ς values) and assigning them similar κ values while making sure that property 1 holds. Also, in both cases, the algorithm guarantees that the minimum κ value allocated is 1, which guarantees that the HMM properties hold (the sum of the emission probabilities for each state is 1 and the sum of the transition probabilities of all the states is 1 [3]).

Algorithm: *calculate-*$\kappa(S_1,S_2)$
Input: Two sets S_1, S_2, $\varsigma(s_i, s_j)$, $\forall\ s_i \in S_1 \wedge s_j \in S_2$
Output: A totally ordered set w: $w \leftarrow((s_i, s_j),\kappa(s_i|s_j))$, $\forall\ s_i \in S_1 \wedge\ s_j \in S_2$
Begin:
Create the set w_s: $w_s \leftarrow((s_i, s_j),\varsigma(s_i, s_j))$, $\forall\ s_i \in S_1 \wedge\ s_j \in S_2$,totally ordered based on $\varsigma(s_i, s_j)$.
Set κ-counter = 1
$\forall w_{s_l} \in w$: Create the set w_{s_l}-*similar* such that:

$$w_i\text{-similar} = \{w_{s_l}\} \cup \{\ j \in w_s \wedge \varsigma(w_{s_l}, s_j) = \varsigma(j, s_j)\}$$

$\forall e \in w_{s_l}$-*similar* :
$\kappa(e|s_j) = \kappa$-counter ++ , if $\varsigma(e, s_j) \neq 0$
$= \infty$, otherwise
Add $((e, s_j), \kappa(e|s_j))$ to w_s
End

4. **Initial κ Values ($\kappa_0(q)$)**
Represents the likelihood of each state being true at the first interval [0,t]. We assign the starting κ-values via a uniform distribution [4]. $\kappa_0(q)$ $\forall q \in Q$ $= k$, where k is randomly generated.

5.2 Experiment: Decoding Spatio-temporal Knowledge

Problem: *Spatio-temporal Decode*

Given a sequence $x = r_1r_2....r_m$ of RCC8 relations, observed at the end of intervals $I_1, I_2, ..., I_m$ respectively, where every interval is of fixed-length Δ.

Find the sequence of motion patterns $p = p_1p_2....p_m$ which maximizes the likelihood of the observed sequence x (P(x,p)), i.e. minimizes $\kappa(x, p)$.

[3] Reminder: In the κ calculus, finding the sum corresponds to obtaining the min.
[4] At the beginning of a stochastic process, all states have equal probabilities of occurrence. This changes as the system progresses through its states.

The solution was implemented directly via the qualitative Viterbi algorithm presented in section 4, by applying it to $\text{HMM}_{st\epsilon}$ of section 5 and testing its results on artificially-generated sequences of RCC8 relations of various lengths.

Running the solution for 30 sequences of various lengths has given a 90-96% accuracy rate, where accuracy is defined as the ratio of the correctly identified most-likely sequence to the total number of sequences of the same length tested.

6 Conclusion and Future Work

We presented a framework for constructing a Qualitative HMM, which uses order-of-magnitude instead of numerical probabilities to capture its transitions, and equipped it with a qualitative equivalent of the Viterbi algorithm for decoding stochastic processes.

We are currently working on equipping our model with the algorithms that enable it to solve the evaluation and learning problems of standard HMMs, and consequently giving it their recognitive and predictive power.

Acknowledgements

The authors would like to acknowledge the support of NSERC through the Discovery Grants program and the support of the Ontario Graduate Scholarship program.

References

1. Bailey-Kellog, C., Zhao, F.: Qualitative spatial reasoning: Extracting and reasoning with spatial aggregates. AI Magazine 24(4), 47–60 (2003)
2. Bicego, M., Grosso, E., Tistarelli, M.: Person authentication from video of faces: A behavioral and physiological approach using pseudo hierarchical hidden markov models. In: ICB, pp. 113–120 (2006)
3. Boxer, P.: Towards learning naive physics by visual observation: Qualitative spatial representations. In: IJCAI, pp. 265–278 (2001)
4. Cohn, A., Hazarika, S.: Qualitative spatial representation and reasoning: An overview. Fundamenta Informatica 46(1-2), 2–32 (2001)
5. Egenhofer, M., Frenzosa, R.: On the equivalence of topological relations. International Journal of Geographical Information Systems 9(2), 133–152 (1995)
6. Freska, C.: Conceptual neighborhood and its role in temporal and spatial reasoning. In: IMACS, pp. 181–187 (1991)
7. Goldszmidt, M., Pearl, J.: Qualitative probabilities for default reasoning, belief revision, and causal modeling. Artificial Intelligence 84, 57–112 (1996)
8. Gooday, J., Cohn, A.: Conceptual neighborhoods in temporal and spatial reasoning. In: ECAI, Workshop on Spatial and Temporal Reasoning (1994)
9. Guesgen, H.: When regions start to move. In: FLAIRS, pp. 465–469 (2003)
10. Hernández, D.: Qualitative Representation of Spatial Knowledge. LNCS, vol. 804. Springer, Heidelberg (1994)

11. Ibrahim, Z., Tawfik, A.: An abstract theory and ontology of motion based on the regions connection calculus. In: SARA' 2007 (to Appear)
12. Karplus, K., Karchin, R., Shackelford, G., Hughey, R.: Calibrating -values for hidden markov models using reverse-sequence null models. Bioinformatics 21(22), 4107–4115 (2005)
13. Köhler, C.: The occlusion calculus. In: Cognitive Vision Workshop (2002)
14. Petrie, T., Baum, L.E.: Statistical inference for probabilistic functions of finite state markov chains. Annals of Mathematics and Statistics 37, 1554–1563 (1966)
15. Lovell, B.: Hidden markov models for spatio-temporal pattern recognition and image segmentation (2001)
16. Rabiner, L.R.: A tutorial on hidden markov models and selected application in speech recognition. Proceedings of IEEE, 77(2) (1989)
17. Randell, D., Cui, Z., Cohn, A.: A spatial logic based on regions and connection. In: KR, pp. 165–176 (1992)
18. Rosti, A., Gales, M.: Factor analysed hidden Markov models for speech recognition. Technical Report 453, Cambridge University Engineering Department (2003)

A Multiobjective Resource-Constrained Project-Scheduling Problem

Fouad Ben Abdelaziz*, Saoussen Krichen, and Olfa Dridi

LARODEC Laboratory, Institut Supérieur de Gestion, University of Tunis, 41 Rue de la liberté, Le Bardo 2000, University of Tunis, Tunisia
fabdelaziz@ud.ac.ae,saoussen.krichen@isg.rnu.tn,dridi_o@yahoo.fr

Abstract. The planning and scheduling activities are viewed profoundly important to generate successful plans and to maximize the utilization of scarce resources. Moreover, real life planning problems often involve several objectives that should be simultaneously optimized and real world environment is usually characterized by uncertain and incontrollable information. Thus, finding feasible and efficient plans is a considerable challenge. In this respect, the Multi-Objective Resource-Constrained Project-Scheduling problem (RCPSP) tries to schedule activities and allocate resources in order to find an efficient course of actions to help the project manager and to optimize several optimization criteria. In this research, we are developing a new method based on Ant System meta-heuristic and multi-objective concepts to raise the issue of the environment uncertainty and to schedule activities. We implemented and ran it on various sizes of the problem. Experimental results show that the CPU time is relatively short. We have also developed a lower bound for each objective in order to measure the degree of correctness of the obtained set of potentially efficient solutions. We have noticed that our set of potentially efficient solutions is comparable with these lower bounds. Thus, the average gap of the generated solutions is not far from the lower bounds.

Keywords: Resource-Constrained Project-Scheduling problem, Ant System, multiobjective optimization.

1 Introduction

The planning and scheduling of activities are important to generate good plans and to maximize the utilization of scarce resources. The planning process consists in generating feasible Course of Actions (COA), the so-called 'plan', that its execution would allow the accomplishment of the tasks or activities. In this respect, the Multi-Objective RCPSP tries to find suitable resources allocation and optimize these objectives. This problem can be stated as a set of jobs or tasks, related by successor and predecessor constraints and where each task requires for its realization a various combination of resources/mode.

* A visiting professor at the school of engineering, American University of Sharjah, UAE.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 719–730, 2007.

The RCPSP has various features as multi-mode, single mode, with non pre-emptive or pre-emptive resources, renewable or non-renewable resources. This problem was been widely studied in the single objective case, however, the multi-objective version of the RCPSP is scant. Several approaches were developed to solve uni-objective RCPSPs as Merkle et al. [8] [7] who applied an Ant System methods to solve the RCPS problem. Demeulemeester et al. [4] applied a Branch-and-Bound procedure for the RCPSP. A Genetic Algorithm has also been applied to solve the RCPSP [10]. For the multi-objective RCPSP, Belfares et al. [3] developed a progressive resource allocation method based on the Tabu Search meta-heuristic. Al-Fawzan and Haouari developed a method based on Tabu Search to solve the RCPSP with two objectives: the makespan and robustness [1].

In this paper, we develop a multi-objective RCPSP with a multi-mode feature and renewable resources. The objectives of the RCPSP are namely: the minimization of the makespan C_{max}, the minimization of the total cost and the maximization of the probability of success.

For that, we develop an Ant System based approach based on multi-objective concepts to schedule the set of tasks while assigning the suitable modes.

Our Multi-Objective Ant System Approach (MOASA) is based on the definition of a prefixed number of ants that operate in terms of the 3 objectives, it iterates until reaching a stable set of potentially set of efficient solution. MOASA was implemented in C++ language and runned on a large test-bed of problems that covers RCPSPs sized from 6 to 250 tasks. We also developed a lower bound for the RCPSP to frame the exact solutions and to evaluate our results. This paper is organized as follows: section 2 describes the RCPSP with multiple objectives and provides its mathematical formulation. Section 3 details the MOASA with all its features. Our approach is illustrated with an example and various computational results. section 4 is devoted to the development of the lower bound and a comparison of the generated solutions and these lower bounds.

2 Problem Statements

The RCPSP can be stated as a set of tasks, related by successor and predecessor constraints and where each task requires for it is realization a various combination of resources, to be assigned to a set of resources of limited capacity, where each resources can be used by various tasks. A resource can be human, material or financial and its availability is uncertain and is considered as a non-deterministic variable. In this respect, the solution is to allocate the necessary and available resources to the tasks over time optimizing several objectives and taking into account contingency aspect of the environment beside the constraints of the problem. The problem can be represented as an oriented graph, with tasks as nodes and precedence constraints as edges, such a graph allows the possibility to specify the combination of resources and the precedence constraint between tasks in figure (1). In this graph, the nodes identify tasks and their resources'combinations and the edges represent the precedence rela-

tionship between them. Indeed, task 2 can be achieved by using the combination of resources R_1 and R_2 or R_3 and R_4. This can only been achieved if task 1 has already been accomplished. The following notation accounts for the mathematical formulation of the RCPSP:

N:	number of tasks.
t_i:	the task i.
k:	number of resources.
M_i:	number of modes that task i can be performed in.
$\|m\|$:	number of resources of the mode m.
PR_i:	the set of predecessors of task i.
PS_i:	the set of successors of task i.
R_j:	the available quantity of resource of type j.
r_j:	the resource of type j.
q_{ijm}:	the quantity of resource of type j required to task i being performed in mode m .
d_{im}:	the duration of task i being performed in mode m .
s_i:	starting time of task i.
l_i:	finishing time of task i with $l_i = s_i + d_{im}$.
a_i^m:	elementary action is the task t_i being realized by the mode m.
COA:	the courses of action $\{a_i^m, i = 1,\ldots,N\}$.
CV_j:	the in-use costs of resource of type j.
c_j:	cost of resource j.
τ_{ij}:	the amount of pheromone where task i is realized by the resource j.
τ:	is the pheromone matrix.
η_{ij}:	is a priori probability of availability of resource j used for task i.
C_{max}:	the makespan $C_{max}=Max\sum_{i=1}^{N} l_i$.
P_{ij}:	the probability that resource j successfully realize task i.
$x_{ijm}=$	$\begin{cases} 1 & \text{if } t_i \text{ is realized by resource } j \text{ using mode } m. \\ 0 & \text{otherwise.} \end{cases}$
f:	number of ants.
PE:	the set of potentially efficient solutions.

The mathematical formulation of the RCPSP is the following:

$$Minimize \;\; C_{max} = Max \sum_{i=1}^{N} l_i \tag{1}$$

$$Minimize \;\; \sum_{i=1}^{N} \sum_{m=1}^{M_i} \sum_{j=1}^{k} c_{ij} \; x_{ijm} \tag{2}$$

$$Maximize \;\; 1/N \sum_{i=1}^{N} (\sum_{m=1}^{M_i} \sum_{j=1}^{k} P_{ij} \; x_{ijm} / \sum_{m=1}^{M_i} \sum_{j=1}^{k} q_{ijm}) \tag{3}$$

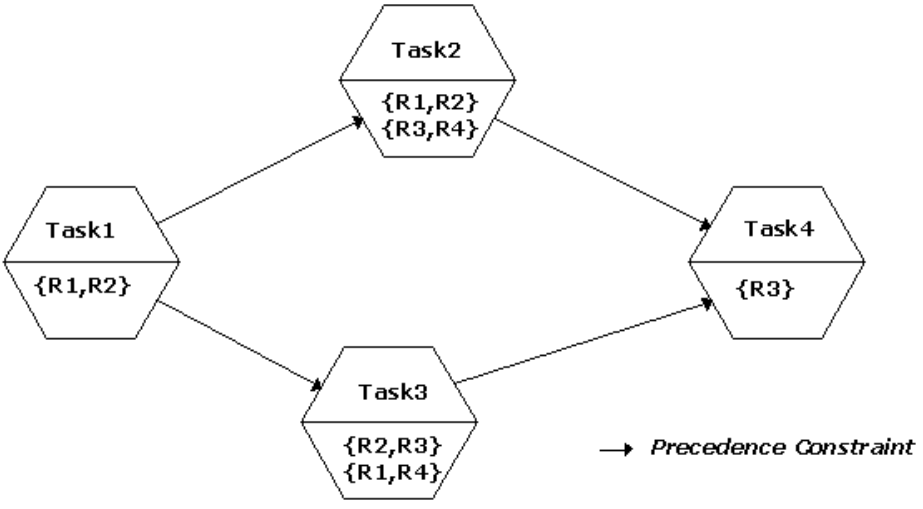


Fig. 1. Structure of the RCPS problem

Subject to:

$$\sum_{i=1}^{N} \sum_{m=1}^{M_i} x_{ijm}\ q_{ijm} \leq R_j, \quad j = 1,\ldots,k. \tag{4}$$

$$s_i \geq Max_{p \in PR_i}(s_p + (\sum_{m=1}^{M_p}(\sum_{j=1}^{k} x_{pjm}/|m|)\ d_{pm})), \quad i = 1,\ldots,N. \tag{5}$$

$$\sum_{m=1}^{M_i} 1/|m| \sum_{j=1}^{k} x_{ijm} = 1, \quad i = 1,\ldots,N. \tag{6}$$

$$x_{ijm} \in \{0,1\}, \quad i = 1,\ldots,N, \quad m = 1,\ldots,M_i, \quad j = 1,\ldots,k. \tag{7}$$

The multi-objective RCPSP consists in optimizing three objectives simultaneously:

- Minimize the makespan C_{max} equation(1);
- Minimize the COA cost equation (2);
- Maximize the probability of success of the COA equation (3).

Under a set of constraints denoted by equations (4-7), the resources availability constraint (4), means that each task i is performed by r_{ik} quantity of resource k that can't exceed the available quantity R_k. Equation (5) is the predecessor constraint. It guarantees that each task i starts if all their predecessors PR_i have been finished. Constraint (6) ensures that each task i is required exactly by one combination of resources. Constraint (7) represents the decision variables, where

$$x_{ij} = \begin{cases} 1 & \text{if resource } j \text{ contributes to the realization of task } i. \\ 0 & \text{otherwise.} \end{cases}$$

A solution to this problem can viewed as sequences of synchronization tasks and resources, where each sequence corresponds to the realization of a given task t_i by a combination of resources c, thus a sequence is an elementary action $a_i^c = (t_i, c)$.

3 An Ant Based Approach to Solve the Multi-objective RCPS Problem

In an attempt to find a sequence of realizable actions that allow an efficient utilization of scarce resources, while optimizing several objectives and predicting unforeseen events, we devolop a new method based on the Ant System metaheuristics to optimize this problem. Thus, we have to solve an *NP-hard* multiobjective optimization problem [6], That's why, an ant algorithm is applied. The parameters of the ant system approach for the RCPSP are adapted as follows:

- **Pheromone trail:** pheromone trail have a high influence during the constructive phase, for that, The trail τ_{ij} corresponds to quantity of trail deposit when task i is executed by resource j.
- **Heuristic information:** η_{ij} is considered as a priori information and it is computed by some heuristic function to indicate the desirabily of moving from state i to state j. In our method, and due to the uncertain aspect of the availability of the resources, the heuristic information indicates the average of the probabilities of disponibility of resources j. Let us illustrate with this example: suppose that the quantity available of resource j is q_1 with probability p_1 and q_2 with probability p_2. Hence, the heuristic information is:
  $\eta_j = ((p_1 * q_1) + (p_2 * q_2))/(q_1 + q_2)$.
- **Transition rule:** is a stochastic search function influences and stimulates the ants'decisions during the construction process. This function is directed (i) by available pheromone trail τ_{ij}, (ii) by a heuristic function η_j and (iii) by a specific data of the problem as the cost of the resource j: c_j and the processing time of the tasks i: d_{im}.

Our method starts with no solution and then each ant builds a solution in n iterations, where at each iteration one task is selected from the list of non-achieved tasks to be realized by the a combination of resources. The choice of the resources is guided and directed by the problem characteristics and the available pheromone trail. And because we have to tackle a multi-objective combinatorial optimization problem we are defined two ant colonies. One of them is about optimizing a single objective: the ACS-time, and the other: ACS-cost tries to optimize multiple objectives. Each ant of the first colony minimizes the total duration of the project. However, the other ants of the second colony move in the sense to search a solution with minimum cost and maximum probability of success. In this respect, each colony has its own pheromone matrix and each ant

of particular colony builds its own solution by using only the pheromone and heuristic informations of its colony. The algorithm can be outlined in terms of the following features :

1. *Initialization phase*: initiate the pheromone matrix. The initial amount of the pheromone is initialized to small positive value τ_0. We have used two pheromone matrices:
- Pheromone matrix task-resource M_I: given n tasks and m resources

$$\tau_{ij} = \begin{cases} \tau_0 & \text{if resource } r_j \text{ contributes to the realization of task } t_i. \\ 0 & \text{otherwise.} \end{cases}$$

$$\tau_{ij} = \begin{pmatrix} \tau_0 & ... & 0 & \tau_0 \\ 0 & \tau_0 & ... & 0 \\ 0 & ... & 0 & \tau_0 \\ \tau_0 & 0 & ... & 0 \end{pmatrix}$$

- Pheromone matrix task-task M_{II}: given n tasks

$$\tau_{iu} = \begin{cases} \tau_0 & \text{if task } t_u \text{ is a floating task.} \\ 0 & \text{otherwise.} \end{cases}$$

$$\tau_{iu} = \begin{pmatrix} 0 & ... & \tau_0 & \tau_0 \\ \tau_0 & 0 & ... & 0 \\ 0 & ... & 0 & \tau_0 \\ \tau_0 & 0 & ... & 0 \end{pmatrix}$$

2. *Construction phase*: the two Ant colonies are activated simultaneously and uses independent pheromone trails:
- the ACS-cost: the goal is to minimize the cost function and to maximize the probability of success. Each ant of the colony uses the pheromone matrix M_I and applies the transition rule P_{ij} to choose the suitable combination of the resources j to be assigned to a given task i. The transition rule of assigning task i to resource j is $P_{ij} = \tau_{ij} * \eta_j / \sum_{l \in \aleph_j} (\tau_{il} * \eta_l)$ where the $\aleph_j$ is neighborhood set of the combination of resources.
- the ACS-time: the goal is to minimize the total duration of the project (the makespan) C_{max}. Each ant chooses randomly the combination of resources to be assigned to a task t_i and simultaneously use the pheromone matrix M_{II} and applies the transition rule P_{iu} to obtain a neighbor by the selection of floating task t_u to be realized at the same period that t_i. Where $P_{iu} = \tau_{iu}/(d_u * \sum_{v \in \aleph_i} \tau_{iv})$ with $\aleph_i$ is the set of neighborhood of taks i.

3. *Updating phase*: the two pheromone matrices are updated locally and globally.
- local update: during the construction phase and after building a solution, the pheromone trail intensity decreases over time to avoid convergence of the algorithm to local optimum and favoring the exploration of not visited areas of the search space. $\tau_{ij} = (1 - \rho)\tau_{ij}$, where ρ is a parameter determine the evaporation rate.
- global update: this procedure is applied at the end of the construction phase. Only the best solutions are allowed to deposit pheromone in order to generate new solutions in the neighborhood of these preferred ones and to favorate the

diversification. $\tau_{ij} = \tau_{ij} + \rho(1/S^*)$, where S^* is the best value for respective matrix and objective.

4. *Filtering process*: this procedure is applied in order to select the potentially efficient solutions. In other words, we eliminate all the dominated solutions and we retain only the non dominated one.

3.1 The Algorithm

- Initialization:
  - Initialize the pheromone matrix M_I
  - Initialize the pheromone matrix M_{II}
  - assigning the ants to the starting task
  - Ψ: set of potentially efficient solutions($\Psi \leftarrow \emptyset$)
  - ψ^t: set of feasible solutions according to the ACS-time ($\psi^t \leftarrow \emptyset$)
  - ψ^c: set of feasible solutions according to the ACS-cost ($\psi^c \leftarrow \emptyset$)
  - m: number of ants of each colony
- Iterative Process:
  at iteration i
  Step 1:
  -for each ant f
  - Perform ACS-cost(f, M_I)
  - Perform ACS-time(f, M_{II})
  - $\psi^c \leftarrow \psi^c \cup \psi^c_i$
  - $\psi^t \leftarrow \psi^t \cup \psi^t_i$
  end for each
  - $\Psi \leftarrow \psi^t \cup \psi^c$
  Step 2:
  - perform global updating
  Step 3:
  - filtrate Ψ in order to have only the set of potentially efficient solutions
  Step 4:
  - If stopping criterion is met, stop
  - Else $i \leftarrow i + 1$ go to step 1.

3.2 An Exemple $n = 6$

We apply The MOASA on a RCPSP with 6 tasks $n = 6$ and 4 resources $k = 4$.

The inputs to the algorithm are reported in table (1) that contains a set of tasks to be accomplish by various combinations of resources. For example, task T_2 can be processed by three possibilities of resources'combinations: the first consists in using 3 unities of R_1 combined with 2 unities of R_3, the second consists in using 3 unities of R_2 combined with 2 unities of R_4 and the thrid consists in using 5 unities of R_4.

Applying the Ant Algorithm to the example in table (1) and varying the number of ants, we obtain the solutions enumerated below:

Table 1. An RCPS problem with $n = 6$ and $k = 4$

Tasks	combinations of resources and quantities		
t_1	$R_1(2)$,$R_2(1)$	$R_4(4)$	-
t_2	$R_1(3)$,$R_3(2)$	$R_2(3)$,$R_4(2)$	$R_4(5)$
t_3	$R_1(2)$,$R_2(1)$	$R_3(3)$	-
t_4	$R_3(2)$,$R_4(2)$	$R_1(4)$	-
t_5	$R_2(3)$,$R_4(1)$	$R_3(2)$	-
t_6	$R_1(1)$,$R_2(1)$	$R_4(2)$	-

number of ants	**CPU time(s)**	**ACS-cost**	**ACS-time**
2	0	$\begin{pmatrix}45\\148\\0.7\end{pmatrix}$	$\begin{pmatrix}30\\150\\0.66\end{pmatrix}$
5	0	$\begin{pmatrix}45\\148\\0.71\end{pmatrix}$	$\begin{pmatrix}30\\150\\0.65\end{pmatrix}$
10	0	$\begin{pmatrix}45\\142\\0.62\end{pmatrix}$	$\begin{pmatrix}30\\148\\0.72\end{pmatrix}$
30	1	$\begin{pmatrix}45\\124\\0.63\end{pmatrix}$	$\begin{pmatrix}30\\150\\0.72\end{pmatrix}$
50	2	$\begin{pmatrix}45\\124\\0.53\end{pmatrix}$	$\begin{pmatrix}30\\150\\0.72\end{pmatrix}$
100	5	$\begin{pmatrix}45\\135\\0.38\end{pmatrix}$	$\begin{pmatrix}30\\150\\0.66\end{pmatrix}$

One of the solution obtained in table $\begin{pmatrix}30\\150\\0.66\end{pmatrix}$ corresponds to ACS-time algorithm.

The total duration of the project (the makespan):

C_{max}= max $\sum_{i=1}^{6} l_i$=30.

The cost of the utilization of the resources by the tasks:

$\sum_{i=1}^{6} \sum_{j=1}^{4} \sum_{m=1}^{M_i} c_{ij}\ x_{ijm} = 150$.

The probability of success:

$1/6 \sum_{i\in 6}(\sum_{m\in M_i} \sum_{j\in 4} P_{ij} x_{ijm} / \sum_{m\in M_i} \sum_{j\in 4} q_{ijm}) = 66\%$

3.3 Case Studies

In this section, we generate a set of case studies by using an algorithm of construction and we give the numerical experiments which are evaluated by our new method as summarized in the following table: in the following tables we report the numerical results provided by our Ant based method:

Problems	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
Number of tasks	6	6	6	6	11	30	50	50	50
Number of resources	3	5	4	5	12	4	3	5	3
Avg.number.predecessors	(1-2)	(1-2)	(1-2)	(2-3)	(1-3)	(1-2)	(1-2)	(1-2)	(2-3)
number of ants	5	5	10	10	10	10	10	10	5
CPU time(s)	0	0	0	1	1	4	5	5	2

Problems	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}	P_{17}	P_{18}
Number of tasks	50	60	100	100	100	100	150	200	250
Number of resources	10	4	3	3	5	15	15	20	15
Avg.number.predecessors	(1-2)	(1-2)	(1-2)	(2-3)	(2-3)	(1-2)	(1-2)	(1-2)	(1-2)
number of ants	10	10	10	5	10	10	10	10	10
CPU time(s)	6	10	19	4	8	8	35	60	59

3.4 A RCPSP with $n = 100$

Numerical experiments have shown that the obtained number of potentially efficient solutions depends on the number of resources.

Tasks	Resources	number of ants	$\|PE\|$	CPU time(s)
100	5	10	4	8
100	15	10	6	9

Generally, when the number of resources rises, the number of resources' combinations rises. Hence, the number of diversified potentially efficient solutions becomes larger.

3.5 A RCPSP with $n = 200$

Let us consider a RCPSP with 200 tasks. When varying the number of resources, we obtain the following table:

k	number of ants	$\|PE\|$	CPU time(s)
1	50	1	914
5	50	9	932
8	50	11	941
15	50	14	895
20	50	16	1070

Thus, the example of $\begin{pmatrix} n = 200 \\ k = 1 \end{pmatrix}$ is a classical RCPS problem that corressponds to the single mode RCPS problem. Each task has only one execution mode for that we have only one potentially efficient solution. However, the examples of $k = \{5, 8, 15, 20\}$ correspond to the multi-mode RCPS problem where each task has several execution mode and where each mode has different value of the cost, the probability of success and the processing time C_{max}. For example, for $k = 5$ we have 9 potentially efficient solutions and for $k = 20$ we have 16 potentially efficient solutions and this due to the increased number of execution mode or resources'combination where the number of resources rises.

4 Lower Bound

Generally, for the *NP*-hard problem it is very difficult to find the optimal solutions. Thus, the solutions obtained from the heuristic methods represents a subset of the optimal ones. Hence, we employed the lower bound (LB) in order to frame the optimal solution of the multi-objective RCPS problem and to calculate the *Gap*: average deviations of the solutions generated by our algorithm from the lower bound value.

$$Gap = (Sol - LB)/LB \tag{8}$$

For that, we establish a lower bound on the makespan and the COA cost objectives denoted respectively by α_1 and α_2.

The lower bound of the makespan α_1 was selected to be the critical path length of the problem, which is equal to the technological earliest completion time [5] [9] and the lower bound of the cost α_2 was obtained by relaxing the resources availability constraint (9):

$$\sum_{i=1}^{N} \sum_{m=1}^{M_i} x_{ijm}\ q_{ijm} \leq R_j, \quad j = 1,\ldots,k. \tag{9}$$

Table 2. Average deviations from the lower bound

Problems	α_1	α_2	Gap_{α_1}	Gap_{α_2}
P_3	8	124	0.12	0
P_5	26	68	0.23	0.13
P_6	56	702	0.08	0.13
P_{10}	44	982	0.27	0.05
P_{11}	115	1594	0.04	0.13
P_{15}	299	2630	0.1	0.14
P_{16}	289	4694	0.09	0.2
P_{17}	395	6357	0.08	0.17
P_{18}	519	6148	0.15	0.17
Avg. *Gap*			0.13	0.12

We drive in the table (2), a series of numerical examples for different RCPSPs. As shown in table (2), for the problem P_3 the Gap_{α_2} is null, so the solution coincides with the lower bound α_2. Hence, our algorithm generate the optimal solution for the COA cost.

The deviation from the LB for the two objectives is relatively small, for the objective α_1 the average Gap is about 0.13 and for α_2 the average Gap is about 0.12. This values are considerably interesting.

5 Conclusion

The multi-objective Resource-Constrained Project Scheduling Problem is an *NP*-hard and hard constrained problem. If we consider several objectives, the problem becomes more complex due to the existence of a set of efficient solutions instead of a single optimal solution. We developed a new approach based on the ant system metaheuristic, denoted by MOASA, in order to generate the set of potentially efficient solutions. We defined a two ant colony system to handle the multiplicity of the objectives. We implemented our algorithm in C++ language and generated the solutions for problems varying from 6 to 250. We explored the RCPSP with multiple objectives through a compuatational experiment. We have also developed lower bounds for the problem and compared the generated results of MOASA to the lower bound in order to measure the degree of correctness of the obtained set of potentially efficient solutions.

References

1. Al-Fawzan, M.A., Haouari, M.: A bi-objective model for robust resource-constrained project scheduling. International Journal of production economics 96, 175–187 (2005)
2. Baar, T., Brucker, P., Knust, S.: Tabu Search Algorithms and Lower Bounds for the Resource-Constrained Project-Scheduling problem. In: Voss, S., Martello, S., Osman, I., Roucailor, C. (eds.) Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, pp. 1–18. Kluwer Academic Publishers, Dordrecht (1999)
3. Belfares, L., Klibi, W., Nassirou, L., Guitouni, A.: Multi-objectives Tabu Search based Algorithm for Progressive Resource Allocation. European Journal of Operational Research 177, 1779–1799 (2007)
4. Demeulemeester, E., Herroelen, W.: A branch and bound procedure for the multiple resource-constrained project-scheduling problems. Management Science 38, 1803–1818 (1992)
5. Kolisch, R., Hartmann, S.: Experimental evaluation of state-of-the-art heuristics for the resource-constrained project Scheduling problem. European Journal of Operational Research 127, 394–407 (2000)
6. Landa Silva, J.D., Burke, B.K., Petrovic, S.: An Introduction to Multiobjective Metaheuristics for Scheduling and Timetabling. Automated Scheduling, Optimization and Planning Research Group School of Computer Science and IT, University of Nottingham, UK (2003)

7. Merkle, D., Middendorf, M.: A New approach to solve permutation scheduling problems with Ant Colony Optimization. In: Applications of Evolutionary Computing: Procceedings of EvoWorkshops, pp. 484–493 (2001)
8. Merkle, D., Middendorf, M., Schmeck, H.: Ant Colony Optimization for Resource-Constrained Project Scheduling. In: Proceedings of the genetic and evolutionary computation conference, pp. 893–900 (2000)
9. Patterson, J.H., Huber, W.D.: A Horizon-Varying, Zero-One Approach to Project Scheduling. Management Science 20(6), 990–998 (1974)
10. Wei Feng, C., Liu, L., Scott Burns, A.: Using Genetic Algorithms to solve Construction time-cost Trade-Off problems. Journal of computing in civil engineering, 184–189 (1997)

Extending Classical Planning to the Multi-agent Case: A Game-Theoretic Approach

Ramzi Ben Larbi, Sébastien Konieczny, and Pierre Marquis

CRIL - CNRS, Université d'Artois - Lens - France

Abstract. When several agents operate in a common environment, their plans may interfere so that the predicted outcome of each plan may be altered, even if it is composed of deterministic actions, only. Most of the multi-agent planning frameworks either view the actions of the other agents as exogeneous events or consider goal sharing cooperative agents. In this paper, we depart from such frameworks and extend the well-known single agent framework for classical planning to a multi-agent one. Focusing on the two agents case, we show how valuable plans can be characterized using game-theoretic notions, especially Nash equilibrium.

1 Introduction

In classical planning, one is interested in computing plans enabling an agent to reach her goals when performed. Among the standard assumptions in classical planning are that the initial state of the world is known by the agent, that each possible action is deterministic and its outcome can be perfectly predicted, that the goals are binary ones (i.e., each state of the world is either a fully satisfactory one or is fully unsatisfactory), and that the world is static in the sense that the only way to change it is to perform one of the agent's actions (thus, not only exogeneous events cannot take place but the world has no intrinsic dynamics).

More sophisticated planning frameworks are obtained by relaxing some of the assumptions above. In particular, in conformant planning, goals are still binary ones and the considered plans are unconditional ones, but it is not assumed that the initial state of the world is fully known or that the available actions are deterministic ones.

In this paper, we extend the classical planning setting to a multi-agent planning one. We consider a group of agents where each agent has its own actions and goals. Agents operate in a common environment. In this new setting, the standard assumptions of classical planning are made. Nevertheless, such asumptions (especially, the static world one and the deterministic actions one) are not enough to allow an agent to predict how the world will evolve after her plan is executed. Indeed, agents' plans interaction introduces some uncertainty. Each agent generally ignores which plans the other agents will point out and how her plan will be interleaved with theirs. We suggest to handle this issue thanks to concepts from game theory; in the new setting we put forward, we show how any agent can achieve a strategic diagnosis of the scenario under consideration, from its game representation.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 731–742, 2007.

Our approach to classical, yet multi-agent planning is meaningful in scenarios where it is not possible to sense/observe during plan execution (a usual assumption in classical planning), or when constraints (on time or ressources) prevent from online re-planning. As a matter of example, consider autonomous high-speed flying robots or infobots working on highly volatile markets. For the sake of simplicity, we focus on the case where each agent knows the goals of each agent from the group, as well as the set of all plans each agent can point out. Those are standard assumptions in game theory. We also assume that the agents have the possibility to coordinate, which means that they can decide to build a common plan. In this case the uncertainty of the execution is removed. Let us consider a (toy) example as an illustration:

Example 1. Two agents, a robot-painter and an robot-electrician, operate in a single room. The bulb has to be changed (which is the goal of the electrician) and the ceiling has to be painted (which is the goal of the painter). The electrician has a new bulb and the painter the materials needed to paint the ceiling. Now, there is a single ladder in the room (the ladder is thus a critical resource). Furthermore, the painter needs some light in the room in order to make her job. The electrician can achieve three actions: TLe ("take the ladder"), CB ("change the bulb"), RLe ("release the ladder"); and the painter three actions: TLp ("take the ladder"), P ("paint"), RLp ("release the ladder") ; P succeeds only if CB has been performed before. TLe and TLp succeed only if the ladder is available (i.e., it has been released before).

The following interactions can be easily envisioned:

- If the painter takes the ladder first, she will not be able to achieve her goal (since the bulb has to be changed first); if she does not release the ladder, then the electrician will not be able to achieve her goal.
- If the electrician takes the ladder first, she will be able to achieve her goal; then, the painter will be able to achieve her goal if and only if the electrician releases the ladder. Accordingly, if both agents coordinate so as to execute the joint plan $\mathsf{TLe.CB.RLe.TLp.P}$, then both agents will be satisfied.

The key questions we address in this paper are the two following ones: for each agent of the group, what are her "best" plans? And does a given plan require coordination to be achieved in a satisfying way? Focusing mainly on the two agents case, we show how a game can be associated to any multi-agent planning problem; accordingly, the "best" plans for a rational agent can be characterized using game-theoretic notions, especially Nash equilibrium. We also identify the scenarios for which a cooperation between agents is likely to occur and show how many strategic information can be derived from the corresponding game. We finally show that several important settings where interacting agents are considered can be cast in our framework, including conformant planning [13,6,7,12] and Boolean games [10,9,8,3].

2 A Framework for Multi-agent Classical Planning

We consider a group of agents $N = \{1, 2, \ldots, k\}$, where each agent is identified by an integer. Let S be a finite (and non-empty) set of (abstract) states. Let us denote by s_0 the initial state, assumed to be the actual state of the world. s_0 is known by each agent from N.

Each agent of N is associated to a finite set of available actions:

Definition 1. ***(action)*** *An* action α *is a mapping from S to S. The set of actions of agent i is denoted by A^i.*

In the following, an action will be denoted by a small Greek letter. Note that the previous definition means that actions are deterministic and fully executable. This last assumption is not very demanding, since if one wants to model that an action is not executable in a state s, then this can be typically represented by an action that does not change the world when performed in s, i.e. $\alpha(s) = s$, or that leads to a "sink" state, i.e., $\alpha(s) = s_\perp$, with $s_\perp$ a non-goal state such that $\beta(s_\perp) = s_\perp$ for every action β.

From her set of available actions, each agent can build some plans:

Definition 2. ***(plan)*** *Let A be a set of actions. A* plan p *on A is denoted by a (possibly empty) sequence of actions of A, i.e., $p = \alpha_1.\alpha_2.\cdots.\alpha_n$, where each $\alpha_i \in A$. Semantically, it is a mapping from S to S, defined from sequential composition of its actions, i.e., for any $s \in S$, $p(s) = s$ if $p = \epsilon$ (the empty sequence), and $p(s) = \alpha_n(\ldots(\alpha_1(s))\ldots)$ otherwise. The set of all plans on A is denoted by A^*.*

Let $p = \alpha_1.\cdots.\alpha_n$ be a plan. A subplan of p is a subsequence of it, i.e., $p' = \alpha'_1.\cdots.\alpha'_m$ is a subplan of p if and only if there exists a strictly increasing mapping t from $\{1, \ldots, m\}$ to $\{1, \ldots, n\}$ s.t. $\forall q \in \{1, \ldots, m\}$, $\alpha'_q = \alpha_{t(q)}$.

Let $p' = \beta_1.\cdots.\beta_r$ be another plan. $p.p'$ denotes the concatenation of p and p', i.e., $p.p' = \alpha_1.\cdots.\alpha_n.\beta_1.\cdots.\beta_r$.

Definition 3. ***(solution plan)*** *Let $G^i \subseteq S$ be the set of goal states for agent i*[1]*. Let $s_0 \in S$ be the initial state. A plan p is a* solution plan *for i iff $p(s_0) \in G^i$.*

In many cases, it is reasonable to assume that only a non-empty subset Π^i of A^{i*} is envisioned by agent i; in particular, due to computational limitations, plans whose length exceeds a given preset bound can be discarded. Nevertheless, it makes sense to assume that Π^i is closed under subplan, i.e., when a plan p belongs to Π^i, then every subplan of it belongs to Π^i as well; in particular, the empty plan ϵ always belongs to Π^i.

We are now ready to define the notions of agent representation and of multi-agent planning problem:

Definition 4. ***(agent representation)*** *Each agent $i \in N$ is characterized by a triple $\mathscr{A}^i = \langle A^i, \Pi^i, G^i \rangle$ consisting of a set of actions A^i, a set of plans $\Pi^i \subseteq A^{i*}$ and a set of goal states G^i.*

[1] We also write $G^i(s) = 1$ when $s \in G^i$ and $G^i(s) = 0$ otherwise.

Definition 5. ***(multi-agent planning problem)*** *A* multi-agent planning problem (MAPP) *for a set N of agents is a triple $\langle S, s_0, \{\mathscr{A}^i \mid i \in N\}\rangle$ consisting of a set of states S, an initial state $s_0 \in S$ and a set of agent's representations $\mathscr{A}^i$ (one per agent).*

When several plans operating on a common environment are furnished (one plan per agent), the final course of events corresponding to their joint execution is one of their shuffles, unless a coordination is achieved. We denote by $\oplus$ the mapping from $A^* \times A^*$ to 2^{A^*} that associates to any pair of plans p_i and p_j, the set containing all their shuffles:

Definition 6. ***(shuffle, shuffle set)*** *Let $p_i = \alpha_1^i.\cdots.\alpha_n^i \in A^{i*}$, $p_j = \alpha_1^j.\cdots.\alpha_p^j \in A^{j*}$. Then $p_i \oplus p_j$ is the set of plans p which are permutations of $p_i.p_j$ for which both p_i and p_j are subplans. Each p is said to be a* shuffle *of p_i and p_j, and $p_i \oplus p_j$ is called the* shuffle set *of p_i and p_j.*

Observe that $\oplus$ is a permutative function (i.e., it is associative and commutative), so the previous definitions of shuffle and shuffle set can readily be extended to the case of more than 2 agents. Observe also that ϵ (the empty sequence) is a neutral element for $\oplus$. Note that such an execution model based on plans shuffling is at work in concrete MAS, like Open Real-Time Strategy Games, see [5].

Example 2. Let us consider again the scenario given in Example 1. Let us call p_1 the robot-electrician plan: TLe.CB and p_2 the robot-painter plan: TLp.P. Then $p_1 \oplus p_2 =$ {TLe.CB.TLp.P, TLe.TLp.CB.P, TLe.TLp.P.CB, TLp.TLe.P.CB, TLp.P.TLe.CB, TLp.TLe.CB.P}.

In the deterministic single agent case, evaluating a plan is quite easy. It is enough to look at the predicted state resulting from the (virtual) execution of the plan: what the agent forsees is what she gets. Characterizing the best plans is an easy task for the agent under consideration: the better the reached state, the better the plan. In the non-deterministic single agent case, the agent has to consider all possible reached states, and to aggregate their scores in order to evaluate a plan (many aggregation functions can be used, e.g. *min* (Wald criterion) for reflecting the behaviour of a pessimistic agent, or using expected utility when the scores are quantitative ones and non-deterministic actions are given by sets of probability distributions).

In the multi-agent (deterministic) case, which is the case we consider in this paper, the situation is similar to the non-deterministic single agent case in the sense that each agent has to consider all possible reached states in order to evaluate her plans. The main difference comes from the nature of uncertainty: in our setting, the uncertainty results from the interaction with the plans furnished by the other agents. Accordingly, each agent has to exploit the fact that she knows the other agents' goals and feasible plans in order to figure out what are her "best" plans. Contrastingly, in the non-deterministic single agent case, the need for handling non-determinism mainly comes from the impossibility to predict in a precise way the result of some actions, like "tossing a coin".

Example 3. If the robot-painter from Example 1 puts forward the plan $p =$ TLp.P.RLp, she is only ensured that the actions of p will be executed in the desired order. While she knows the electrician representation, she does not know which plan the electrician will choose (indeed, the set of feasible plans is not a singleton in general). Even if this set is a singleton, the painter still ignores the execution ordering, i.e., how her plan will interact with the electrician's one. Suppose that the electrician puts forward the plan $p' =$ TLe.CB.RLe. The joint plan that will be finally executed can be any plan from $p \oplus p'$. The resulting uncertainty dissapears whenever the two agents coordinate to put forward a common plan $p''=$ TLp.P.RLp.TLe.CB.RLe.

In our setting, a key issue for each agent is to evaluate the interaction of her plans with the plans of the other agents. Formally, this calls for an evaluation of each shuffle set. To this purpose, we define the notion of satisfaction profile (SP), which is an abstract, summarized, view of shuffle sets evaluation for all the agents of the group. Let us explain how we construct a SP in the two agents situation. Given a pair of plans $p_i \in \Pi^i$ and $p_j \in \Pi^j$, each shuffle from the shuffle set $p_i \oplus p_j$ is a plan built from the actions of both agents; the execution of such a plan leads to a specific final state which is more or less satisfactory for each agent. The evaluation of a plan depends on the state resulting from its execution. We can depict the evaluation of this shuffle set by agent i using a 2-axis representation associating a dot on coordinate (x,y) to a shuffle p iff $G^i(p(s_0)) = x$ and $G^j(p(s_0)) = y$. Note that such a representation can be easily generalised to a n-player situation.

Definition 7. ***(satisfaction profile)*** *Given a MAPP for a set $N = \{1, \ldots, m\}$ of agents, with an initial state s_0, a* satisfaction profile (SP) *for the shuffle set $p_1 \oplus p_2 \oplus \ldots \oplus p_m$ where each $p_i \in \Pi^i$ (with $i \in \{1, \ldots, m\}$) is a set $SP(p_1 \oplus p_2 \oplus \ldots \oplus p_m)$ of vectors $(x_1, \ldots, x_m)$ such that $(x_1, \ldots, x_m) \in SP(p_1 \oplus p_2 \oplus \ldots \oplus p_m)$ if and only if $\exists p \in p_1 \oplus p_2 \oplus \ldots \oplus p_m$ such that for all $i \in \{1, \ldots, m\}$, $G^i(p(s_0)) = x_i$.*

When we consider only two agents i and j, the set of all possible SPs is given on Figure 1.

Numerous conclusions can be drawn from such SPs. Thus, some SPs are clearly better for an agent than other ones. Clearly, SP 2, where all shuffles lead to states that agent i evaluates to 1, is more interesting for her than SP 10, where all shuffles lead to non-goal states (i.e., states that agent i evaluates to 0). Let us also consider SP 3: for each of the two agents, at least one shuffle leads to a bad state (i.e., a non-goal state), and at least one shuffle leads to a goal state. This SP also shows the existence of at least one win-win shuffle (leading to the $(1, 1)$ vector). In such a case, if both agents are rational ones (i.e., they act so as to make the world change to a goal state), then they have to coordinate. Indeed, coordination is a way to get rid of uncertainty. If the two agents i and j put forward two plans $p_i \in \Pi^i$ and $p_j \in \Pi^j$ in an independent way, they risk that the joint execution from $p_i \oplus p_j$ leads to a state evaluated as $(1, 0)$ or as $(0, 1)$, in which case one of the two agents will be unsatisfied. Contrastingly, if they coordinate and jointly put forward a plan corresponding to a win-win shuffle,

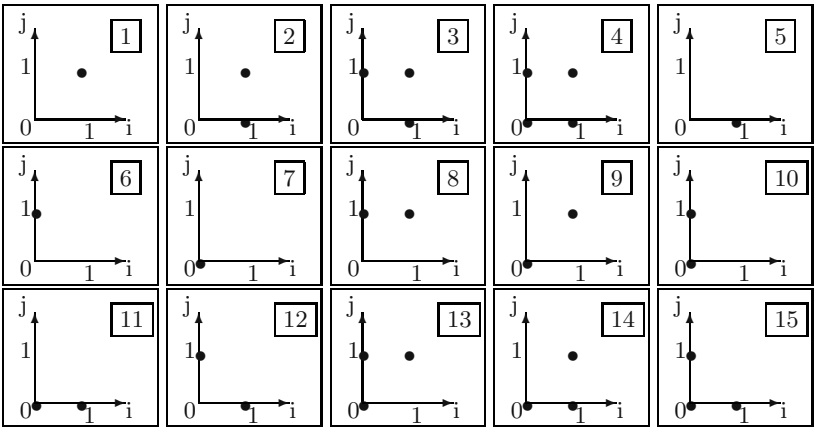


Fig. 1. All possible SPs

they are guaranteed to be both satisfied. So, when a shuffle leads to SP 3, both agents have interest in offering (and accepting) a coordination.

In absence of further information (especially, a probability distribution on the shuffle set), it makes sense to classify all the SPs w.r.t. an ordered scale reflecting the agent's opportunities. Let us take the point of view of agent i and show how the SPs can be gathered and ordered:

Always Satisfied. SPs $1, 2, 5$. For all those SPs, agent i is ensured to reach her goals even if agent j does not accept any coordination. This is the most favourable case for agent i.

Mutual Interest. SPs $3, 4, 9, 13, 14$. For any of those SPs, some joint executions do well and others do bad (for both agents), but they all share the $(1, 1)$ vector, meaning that if the two agents coordinate, they can both reach their goals.

Dependence. SPs $8, 11$. For those SPs, the evaluation of the shuffle set for the other agent does not depend on the joint execution. This means that, *a priori*, there is no "objective" reason for the other agent to accept/decline a coordination in order to help agent i to reach her goal.

Antagonism. SPs $12, 15$. Those SPs reflect more problematic scenarios than the previous ones since the interests of the two agents are clearly distinct. This means that if one is satisfied, then the other one is not (in particular the coordination $(1, 1)$ is never an option). In such cases, agent i can just hope that the joint execution will be good for her.

Always Dissatisfied. SPs $6, 7, 10$. In every course of event, agent i will be dissatisfied (no joint execution allows the agent's goals to be reached). Such SPs are clearly the worst ones for agent i.

Our claim is that, in absence of further information, such a classification is the most rational one. Hence, we consider that each agent i has the following preferences on the evaluations of shuffle sets:

Always Satisfied > **Mutual Interest** >
Dependence > **Antagonism** > **Always Dissatisfied**

where $X > Y$ meaning that SPs of class X are strictly prefered to SPs of class Y, and that all SPs in a given class are indifferent. We can easily encode such a total pre-order in a concise way, using numbers. Thus, we write $e_i(p_i \oplus p_j) = 4$ if and only if $SP(p_i \oplus p_j) \in$ **Always Satisfied**(i), ..., $e_i(p_i \oplus p_j) = 0$ if and only if $SP(p_i \oplus p_j) \in$ **Always Dissatisfied**(i) (see Table 1).

Table 1. SPs evaluation

Class	Evaluation
Always Satisfied	4
Mutual Interest	3
Dependence	2
Antagonism	1
Always Dissatisfied	0

Such evaluations $e_i(p_i \oplus p_j)$ can be roughly seen as utilities, but they do not depend solely on the goals of agent i. Note also that the exact numbers that are used are not really important, just the order matters (our setting is not quantitative at all).

Note finally that, while the definitions to come will use those evaluations $e_i(p_i \oplus p_j)$ and $e_j(p_i \oplus p_j)$, such definitions are still meaningful when other evaluations are used. Thus, if one disagrees with the proposed scale, the following definitions still apply (as soon as all the possible pairs of plans can be evaluated and totally ordered by the agents).

3 Solving the Game and Generating Strategic Diagnoses

From the previous construction we are now able to associate to each shuffle set an evaluation for each agent. This allows us to model the interaction between agents' plans as a game in strategic form. Note that extensive form game cannot work here since it cannot handle the shuffle situation (more exactly, it would lead to awful games in extensive form since there are too many possibilities).

Indeed, to each MAPP for a set of two agents $N = \{1, 2\}$, one can associate a game in strategic form, defined by the set N of players, the set of strategies for each player (the sets Π^1 and Π^2 of plans in our case), and by an evaluation function for each player that associates an evaluation to each profile of strategies (the evaluations $e_1(p_1 \oplus p_2)$ and $e_2(p_1 \oplus p_2)$ of each shuffle set $p_1 \oplus p_2$ in our case).

Example 4. Let us consider the following MAPP: $\langle S, s_0, \{\mathscr{A}^i \mid i \in \{1, 2\}\}\rangle$. $\mathscr{A}^1 = \langle A^1, \Pi^1 = \{p_1, p'_1\}, G^1\rangle$. $\mathscr{A}^2 = \langle A^2, \Pi^2 = \{p_2, p'_2\}, G^2\rangle$. Suppose that the obtained SPs are the ones given in Figure 2.

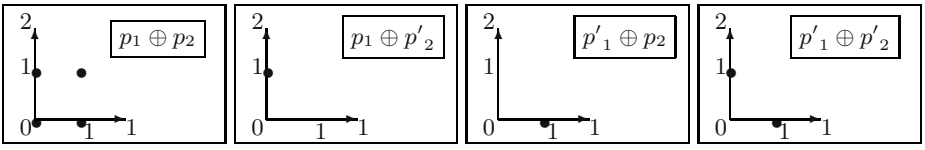


Fig. 2. Example of SPs

We can now associate to this MAPP the game in strategic form given in Table 2.

Table 2. Associated game

	p_2	p'_2
p_1	(3,3)	(0,4)
p'_1	(4,0)	(1,1)

In such a setting there are several candidates for the notion of "best plan". One of them is based on the security level of the plans.[2]

Definition 8. ***(security level of a plan)*** *Given a MAPP for $N = \{1, 2\}$, the* security level *of a plan p_i of an agent i ($i \in N$) facing the set Π^j of plans of agent j ($j \neq i$) is defined as the minimum evaluation of the shuffle set between plan p_i and a plan of player j, i.e.,*

$$S_{\Pi^j}(p_i) = \min_{p_j \in \Pi^j} e_i(p_i \oplus p_j).$$

From the security levels of plans of an agent one can define the security level of the agent:

Definition 9. ***(security level of an agent)*** *Given a MAPP for $N = \{1, 2\}$, the* security level *of agent i facing the set Π^j of plans of agent j, is the greatest security level of agent i's plans, i.e.,*

$$S_{\Pi^j}(i) = \max_{p_i \in \Pi^i} S_{\Pi^j}(p_i).$$

A solution of the game associated to a given MAPP can be defined as a pair of plans $\langle p_1 \in \Pi^1, p_2 \in \Pi^2 \rangle$ such that p_1 (resp. p_2) maximizes the security level of agent 1 (resp. 2) facing Π^2 (resp. Π^1).

Such a notion of solution makes sense in our framework since it can be roughly seen as a worst case analysis of the strategic interaction. Indeed, SPs are a (summarized view of the) set of possible results, and as the SP classification we

[2] While we focus on the 2-agent case, the following notions can be straightforwardly extended to the n-agent case.

pointed out mainly relies on worst case analysis, it is sensible to use security levels as well to compare shuffles.

Nevertheless, security levels do not take into account all agents' opportunities. A much more widely accepted concept of solution is Nash equilibrium [11].

Definition 10. ***(Nash equilibrium)*** *Given a MAPP for $N = \{1, 2\}$, a pair of plans $\langle p_1 \in \Pi^1, p_2 \in \Pi^2 \rangle$ is a Nash equilibrium if none of the agents can get a better evaluation by choosing another plan, i.e., $\langle p_1, p_2 \rangle$ is a Nash equilibrium if and only if $\nexists p \in \Pi^1$ s.t $e_1(p \oplus p_2) > e_1(p_1 \oplus p_2)$ and $\nexists p \in \Pi^2$ s.t. $e_2(p_1 \oplus p) > e_2(p_1 \oplus p_2)$.*

Example 5. Let us step back to the game given in Table 2. And let us consider the pair $\langle p'_1, p'_2 \rangle$. Agent 1 has no incentive to deviate alone from this pair. Indeed, $\langle p_1, p'_2 \rangle$ leads her to a less favorable situation ($e_1(p_1 \oplus p'_2) < e_1(p'_1 \oplus p'_2)$). Similarly, $\langle p'_1, p_2 \rangle$ is clearly less profitable to agent 2 than $\langle p'_1, p'_2 \rangle$. Thus, we can conclude that $\langle p'_1, p'_2 \rangle$ is a Nash equilibrium. It is easy to check that it is the only Nash equilibrium of this game.

In our setting, as in the general case in game theory, it may happen that no Nash equilibrium (in pure strategies) exists, or that several Nash equilibria exist. When there are several Nash equilibria, other criteria, such as Pareto optimality,[3] can be used so as to discriminate them further. The following propositions give two sufficient conditions for the existence of such equilibria.

Proposition 1. *Let us consider a MAPP for two agents* 1 *and* 2 *such that $G^1 = G^2$. Then the associated game exhibits a Nash equilibrium.*

In particular, if the agents share the same goals and if there exists a joint plan that can achieve one of these goals, then our model will point it out as a solution.

Proposition 2. *Let us consider a MAPP for two agents* 1 *and* 2*. Let us denote by $G^{1,+}$ (resp. $G^{2,+}$) the subset of G^1 (resp. G^2) of states reachable using plans on A^1 (resp. A^2) and by $G^{1,2,+}$ (resp. $G^{2,1,+}$) the subset of G^1 (resp. G^2) of states reachable using plans on $A^1 \cup A^2$. If $G^{1,+} = G^{2,+} = \emptyset$ and $G^{1,2,+} = G^{2,1,+} \neq \emptyset$, then the game associated to MAPP exhibits a Nash equilibrium.*

Note that, in our setting, the "prisoner's dilemma" situation, a particular game situation widely studied (see e.g. [1,2]), can also be reached. Like in Example 4 (see Table 2): $\langle p'_1, p'_2 \rangle$ is a Nash equilibrium, but the pair $\langle p_1, p_2 \rangle$ which Pareto-dominates (i.e. is more profitable for both agents than) $\langle p'_1, p'_2 \rangle$ is not a Nash equilibrium (so each agent is tempted to use the other plan).

Interestingly, each of the two agents i and j involved in the MAPP under consideration can derive a number of strategic information from the corresponding game. Due to space limitations, let us only focus on the notions of robust plan, synergetic effect and independence, successively

[3] A vector Pareto-dominates another one if each of the components of the first one is greater or equal to the corresponding component in the second one.

- A plan p_i for agent i is *robust* with respect to agent j if and only if its joint execution with any plan from agent j is ensured to reach the goals of i. In the game in strategic form, such a plan corresponds to a row (or a column) for which all the evaluations for this agent are 4: $\forall p_j \in \Pi^j, e_i(p_i \oplus p_j) = 4$. Clearly enough, such a plan maximizes the security level of agent i. If a robust plan exists for agent i, then no coordination is needed with agent j.
- The existence of a synergy between the two agents can also be easily derived from the game in strategic form. Indeed, a *synergetic effect* for agents i and j is possible if and only if there exist $p_i \in \Pi^i$ and $p_j \in \Pi^j$ such that $e_i(p_i \oplus p_j) > max_{p \in \Pi^i} e_i(\{p\})$ and $e_j(p_i \oplus p_j) > max_{p \in \Pi^j} e_j(\{p\})$. Clearly enough, no synergetic effect is possible when at least one of the two agents has a robust plan.
- A notion of independence between agents, reflecting the fact that no interaction occurs, can also be easily derived from the game in strategic form. Indeed, the two agents are *independent* if and only if $\forall p_i \in \Pi^i, \forall p_j \in \Pi^j, e_i(p_i \oplus p_j) = e_i(\{p_i\})$ and $e_j(p_i \oplus p_j) = e_j(\{p_j\})$.

4 Generality of the Framework

4.1 Conformant Planning

In conformant planning (see e.g. [13,6,7,12]), one is interested in determining whether a sequence of actions (i.e., a plan) is robust (or conformant), i.e., whether it will achieve the goal for all possible contingencies.

Definition 11. ***(conformant planning)***

- *A* non-deterministic action α *over a finite and non-empty set* S *of states is a mapping from* S *to* $2^S \setminus \{\emptyset\}$.
- *A* non-deterministic plan π *on a set* A *of non-deterministic actions (over* S*) is a finite sequence of elements of* A.
- *A* trajectory *for a non-deterministic plan* $\pi = \alpha_1.\cdots.\alpha_n$ *given an initial state* $s_0 \in S$ *is a sequence of states* $s_0, \ldots, s_{n+1}$ *s.t. for every* $i \in 0 \ldots n$, $s_{i+1} \in \alpha_i(s_i)$.
- *A non-deterministic plan* $\pi = \alpha_1.\cdots.\alpha_n$ *on* A *is* conformant *for a goal* $G \subseteq S$ *given an initial state* $s_0 \in S$ *if and only if for every trajectory* $s_0, \ldots, s_{n+1}$ *for* π, $s_{n+1} \in G$.

This problem can be easily cast into our framework. The key idea is to consider every possible trajectory attached to a non-deterministic plan as the result of a possible shuffle with a plan supplied by a second agent who plays the role of Mother Nature; consider the first action α of the plan and assume it has at most k possible outcomes. In this case, the second agent's plan will start with actions $\alpha'_1, ..., \alpha'_k$ wher each α'_j is the void action if α has not been executed (which is encoded using a specific fluent) and achieves the j^{th} outcome of α otherwise. It mainly remains to repeat it for every action of the first plan and to update the second agent's plan by concatening it with the subplan obtained at each step.

4.2 Boolean Games

Definition 12. ***(Boolean game)****[10,9,8,3] A* Boolean game *is a 4-tuple* $G = \langle N, V, \Pi, \Phi \rangle$ *where* $N = \{1, \cdots, n\}$ *is a set of agents,* V *is a set of propositional variables (decision variables),* $\Pi : N \rightarrow 2^V$ *a control assignment function that induces a partition* $\{\pi_1, \cdots, \pi_n\}$ *of* V *with* π_i *the set of variables controlled by agent* i*,* $\Phi = \{\phi_1, \cdots, \phi_n\}$ *a set of formulas.*

For a player $i \in N$, a strategy is a truth assignment of her controlled variables (i.e., a mapping from $\Pi(i)$ to $\{0, 1\}$). A strategy profile consists of the assignments of all the considered agents, and can be viewed as a truth assignment on V (i.e., a mapping from V to $\{0, 1\}$). Agent i is satisfied by a strategy profile P if and only if P is a model of ϕ_i.

We can cast this framework into our one by associating to each variable $v \in V$ a deterministic action v^+ which sets variable v to 1. To each boolean game $G = \langle N, V, \Pi, \Phi \rangle$ we can associate a MAPP $\langle S, s_0, \{\mathscr{A}^i \mid i \in N\} \rangle$ where S is the set of all truth assignments on V, s_0 is the truth assignment s.t. $s_0(v) = 0$ for all $v \in V$. For every agent i, $A^i = \{v^+ \mid v \in \pi_i\}$, Π^i is the subset of plans from A^{i*} such that every action has at most one occurrence in each plan and G^i is the set of models of ϕ_i.

5 Related Work and Conclusion

While much work has been devoted for the past few years to multi-agent planning, they typically assume that agents share some common goals. Relaxing this assumption has a major impact on the possible approaches to tackle the problem and calls for game-theoretic notions.

A closely related approach to our own one is described in [4]. In this paper, policies at the group level are evaluated w.r.t. each agent and the "best ones" are characterized as Nash equilibria, as it is the case in our work. The approach nevertheless departs from our own one by a number of aspects:

- The framework under consideration is planning under uncertainty with full observability and not classical planning. Non-deterministic actions are considered and a set of possible initial states (and not a single state) is known by each agent. Policies are mappings associating actions to states and not linear plans (sequences of actions), and the quality of a plan is not binary in essence (contrariwise to what happens in the classical framework).
- Policies at the group level are part of the input and policies at the agent level are not (while possible plans at the group level are characterized as shuffles from plans at the agent level in our setting).
- Finally, no notion of strategical diagnosis is considered (especially, the need for coordination cannot be derived from the input since policies at the agent level are not relevant).

In this work we have proposed a framework to model multi-agent planning problems. This framework allows to draw strategic conclusions about specific

interactions, and also allows to *solve* many situations. A main point is to show how each MAPP into consideration can be associated to a suitable representation (SP) which can be evaluated (as a number), and this allows for exploiting easily notions and results from game theory. As far as we know, there is no similar notions of SP and evaluations in the literature. Representation and algorithmic aspects are issues for further research.

References

1. Axelrod, R.: The Evolution of Cooperation. Basic Books, New York (1984)
2. Beaufils, B., Delahaye, J.P., Mathieu, P.: Complete classes of strategies for the classical iterated prisoner's dilemma. In: Proc. of EP'98, pp. 33–41 (1998)
3. Bonzon, E., Lagasquie-Schiex, M., Lang, J., Zanuttini, B.: Boolean games revisited. In: Proc. of ECAI'06, pp. 265–269 (2006)
4. Bowling, M., Jensen, R., Veloso, M.: A formalization of equilibria for multiagent planning. In: Proc. of IJCAI'03, pp. 1460–1462 (2003)
5. Buro, M.: Real-time strategy games: A new AI research challenge. In: Proc. of IJCAI'03, pp. 1534–1535 (2003)
6. Cimatti, A., Roveri, M.: Conformant planning via model checking. In: Proc. of ECP'99, pp. 21–34 (1999)
7. Cimatti, A., Roveri, M., Bertoli, P.: Conformant planning via symbolic model checking and heuristic search. Artificial Intelligence 159(1-2), 127–206 (2004)
8. Dunne, P.E., van der Hoek, W.: Representation and complexity in boolean games. In: Alferes, J.J., Leite, J.A. (eds.) JELIA 2004. LNCS (LNAI), vol. 3229, pp. 347–359. Springer, Heidelberg (2004)
9. Harrenstein, P.: Logic in Conflict. PhD thesis, Utrecht University (2004)
10. Harrenstein, P., van der Hoek, W., Meyer, J., Witteveen, C.: Boolean games. In: Proc. of TARK'01, pp. 287–298 (2001)
11. Nash, J.: Equilibrium points in N-person games. Proceedings of the National Academy of Sciences of the United States of America 36(1), 48–49 (1950)
12. Palacios, H., Geffner, H.: Compiling uncertainty away: Solving conformant planning problems using a classical planner (sometimes). In: Proc. of AAAI'06 (2006)
13. Son, T.C., Tu, P.H., Gelfond, M., Morales, A.R.: Conformant planning for domains with constraints-a new approach. In: Proc. of AAAI'05, pp. 1211–1216 (2005)

Dependencies Between Players in Boolean Games

Elise Bonzon, Marie-Christine Lagasquie-Schiex, and Jérôme Lang

IRIT, UPS, F-31062 Toulouse Cédex 9, France
{bonzon,lagasq,lang}@irit.fr

Abstract. Boolean games are a logical setting for representing static games in a succinct way, taking advantage of the expressive power and conciseness of propositional logic. A Boolean game consists of a set of players, each of them controls a set of propositional variables and has a specific goal expressed by a propositional formula. There is a lot of graphical structures hidden in a Boolean game: the satisfaction of each player's goal depends on players whose actions have an influence on these goals. Even if these dependencies are not specific to Boolean games, in this particular setting they give a way of finding simple characterizations of Nash equilibria and computing them.

1 Introduction

The framework of Boolean games [1,2,3,4] allows for expressing compactly static games with binary preferences: each player of a Boolean game controls a set of propositional variables, and a player's preferences is expressed by a plain propositional formula.[1] Thus, a player in a Boolean game has a dichotomous preference relation: either her goal is satisfied or it is not. This restriction may appear at first glance unreasonable. However, many concrete situations can be modelled as games where agents have dichotomous preferences. Furthermore, Boolean games can easily be extended to allow for non-dichotomous preferences, represented in some compact language for preference representation (see [5]).

Using the syntactical nature of goals, we can represent graphically the dependencies between players: if the goal (and hence the satisfaction) of a player i depends on some variables controlled by a player j, then i may need some action of j to see her goal satisfied. This dependency betweeen players is a central notion in *graphical games* [6,7] as well as in [8] – see Section 6. Representing these dependencies on a graph will allow us to compute pure-strategy Nash equilibria in a much simpler way, without enumerating all combinations of strategies.

After recalling some background on Boolean games in Section 2, we introduce in Section 3 the notion of dependency graph between players in Boolean games. In Section 4 we show how this graph can be exploited so as to find simple characterizations Nash equilibria in Boolean games, and we generalize some of these results for non-dichotomous preferences in Section 5. Related work and further issues are discussed in Section 6.

[1] We refer here to the version of Boolean games defined in [4], which generalizes the initial proposal [1].

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 743–754, 2007.

2 *n*-Players Boolean Games

For any finite set $V = \{a, b, \ldots\}$ of propositional variables, L_V denotes the propositional language built up from V, the Boolean constants $\top$ and $\bot$, and the usual connectives. Formulas of L_V are denoted by φ, ψ etc. A *literal* is a variable x of V or the negation of a variable. A *term* is a consistent conjunction of literals. A *clause* is a disjunction of literals. If α is a term, then $Lit(\alpha)$ is the set of literals appearing in α. If $\varphi \in L_V$, then $Var(\varphi)$ denotes the set of propositional variables appearing in φ.

2^V is the set of the interpretations for V, with the usual convention that for $M \in 2^V$ and $x \in V$, M gives the value *true* to x if $x \in M$ and *false* otherwise. $\models$ denotes the consequence relation of classical propositional logic.

Let $V' \subseteq V$. A V'-interpretation[2], also said partial interpretation, is a truth assignement to each variable of V', that is, an element of $2^{V'}$. V'- interpretations are denoted by listing all variables of V', with a $^-$ symbol when the variable is set to false: for instance, let $V' = \{a, b, d\}$, then the V'-interpretation $M = \{a, d\}$ assigning a and d to true and b to false is denoted by $a\overline{b}d$. $\subseteq X$, then

If $\{V_1, \ldots, V_p\}$ is a partition of V and $\{M_1, \ldots, M_p\}$ are partial interpretations, where $M_i \in 2^{V_i}$, $(M_1, \ldots, M_p)$ denotes the interpretation $M_1 \cup \ldots \cup M_p$.

Finally, we denote the partial instantiation of a formula φ by an X-interpretation M_X by: $(\varphi)_{M_X} = \varphi_{v \in M_X \leftarrow \top, v \in X \setminus M_X \leftarrow \bot}$.

Given a set of propositional variables V, a Boolean game on V is a n-players game[3], where the actions available to each player consist in assigning a truth value to each variable in a given subset of V. The preferences of each player i are represented by a propositional formula φ_i formed upon the variables in V.

Definition 1. *A* ***n*-player Boolean game** *is a 4-tuple* (N, V, π, Φ)*, where* $N = \{1, 2, \ldots, n\}$ *is a finite set of players (also called agents);* V *is a finite set of propositional variables;* $\pi : N \mapsto 2^V$ *is a control assignment function;* $\Phi = \{\varphi_1, \ldots, \varphi_n\}$ *is a set of goals, where each* φ_i *is a satisfiable formula of* L_V.

The control assignment function π maps each player to the variables she controls. For the ease of notation, the set of all the variables controlled by i is written π_i instead of $\pi(i)$. Each variable is controlled by one and only one agent, that is, $\{\pi_1, \ldots, \pi_n\}$ forms a partition of V.

Definition 2. *Let* $G = (N, V, \pi, \Phi)$ *be a Boolean game. A* **strategy** *for player i in G is a* π_i*-interpretation . The set of strategies for player i in G is* $S_i = 2^{\pi_i}$ *A* **strategy profile** *s for G is a n-uple* $s = (s_1, s_2, \ldots, s_n)$ *where for all i,* $s_i \in S_i$*.* $S = S_1 \times \ldots \times S_n$ *is the set of all strategy profiles.*

Note that since $\{\pi_1, \ldots, \pi_n\}$ forms a partition of V, a strategy profile s is an interpretation for V, i.e., $s \in 2^V$. The following notations are usual in game theory. Let

[2] Note that a V-interpretation is an interpretation.

[3] In the original proposal [1], Boolean games are two-players zero-sum games. However the model can easily be generalized to n players and non necessarily zero-sum games [4].

$s = (s_1, \ldots, s_n)$ be a strategy profile. For any non empty set of players $I \subseteq N$, the projection of s on I is defined by $s_I = (s_i)_{i \in I}$ and $s_{-I} = s_{N \setminus I}$. If $I = \{i\}$, we denote the projection of s on $\{i\}$ by s_i instead of $s_{\{i\}}$; similarly, we note s_{-i} instead of $s_{-\{i\}}$. π_I denotes the set of the variables controlled by I, and $\pi_{-I} = \pi_{N \setminus I}$. The set of strategies for $I \subseteq N$ is $S_I = \times_{i \in I} S_i$. If s and s' are two strategy profiles, (s_{-I}, s'_I) denotes the strategy profile obtained from s by replacing s_i with s'_i for all $i \in I$.

The goal φ_i of player i is a compact representation of a dichotomous preference relation, or equivalently, of a binary utility function $u_i : S \to \{0,1\}$ defined by $u_i(s) = 0$ if $s \models \neg\varphi_i$ and $u_i(s) = 1$ if $s \models \varphi_i$. s is at least as good as s' for i, denoted by $s \succeq_i s'$, if $u_i(s) \geq u_i(s')$, or equivalently, if $s \models \neg\varphi_i$ implies $s' \models \neg\varphi_i$; s is strictly better than s' for i, denoted by $s \succ_i s'$, if $u_i(s) > u_i(s')$, or, equivalently, $s \models \varphi_i$ and $s' \models \neg\varphi_i$.

This choice of binary utilities implies a loss of generality, even if some interesting problems have naturally dichotomous preferences. We relax this assumption in Section 5, where we consider generalized Boolean games with nondichotomous preferences expressed in some logical language for compact preference representation, as in [5].

3 Dependencies Between Players

We now focus on the syntactical nature of goals, which may help us identifying some game-theoretical notions, as pure-strategy Nash equilibria. Intuitively, if the goal φ_i of player i does not involve any variable controlled by player j, then the satisfaction of i does not depend directly on j. This is only a sufficient condition: it may be the case that the syntactical expression of φ_i mentions a variable controlled by j, but that this variable plays no role whatsoever in the satisfaction of φ_i, as variable y in $\varphi_i = x \wedge (y \vee \neg y)$. We therefore use a stronger notion of formula-variable independence [9].

Definition 3. *A propositional formula φ is* **independent from** *a propositional variable x if there exists a formula ψ logically equivalent to φ and in which x does not appear.*[4]

Definition 4. *Let $G = (N, V, \pi, \Phi)$ be a Boolean game. The set of* **relevant variables** *for a player i, denoted by $RV_G(i)$, is the set of all variables $v \in V$ such that φ_i is not independent from v.*

For the sake of notation, the set of relevant variables for a given Boolean game G will be denoted by RV_i instead of $RV_G(i)$. We now easily define the *relevant players* for a given player i as the set of players controlling at least one variable of RV_i.

Definition 5. *Let $G = (N, V, \pi, \Phi)$ be a Boolean game. The set of* **relevant players** *for a player i, denoted by RP_i, is the set of agents $j \in N$ such that j controls at least one relevant variable of i: $RP_i = \bigcup_{v \in RV_i} \pi^{-1}(v)$*[5].

[4] We have this equivalent semantical characterization of formula-variable independence [9]: φ is independent from x if there exists an interpretation s such that $s \models \varphi$ and switch$(s,x) \models \varphi$, where switch(s,x) is obtained by switching the value of x in s, and leaving the values of other variables unchanged.

[5] Again, the set of relevant players for a Boolean game G should be denoted by $RP_G(i)$: for the ease of notation we simply write RP_i.

Example 1. 3 *friends (*1, 2 *and* 3*) are invited at a party.* 1 *wants to go at this party.* 2 *wants to go at the party if and only if* 1 *goes, whereas* 3 *wants to go there, and prefers that* 2 *goes to, and* 1 *doesn't. This situation can be modelled by the following Boolean game* $G = (N, V, \pi, \Phi)$*, defined by* $V = \{a, b, c\}$*, with* a *means "*1 *goes at the party", the same for* b *and* 2*; and for* c *and* 3*;* $N = \{1, 2, 3\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\varphi_1 = a$, $\varphi_2 = a \leftrightarrow b$ *and* $\varphi_3 = \neg a \wedge b \wedge c$.

We can see that 1*'s satisfaction depends only on herself, 2's depends on 1 and herself, whereas 3's depends on 1, 2 and herself. So, we have:* $RV_1 = \{a\}$, $RV_2 = \{a, b\}$, $RV_3 = \{a, b, c\}$, $RP_1 = \{1\}$, $RP_2 = \{1, 2\}$, $RP_3 = \{1, 2, 3\}$.

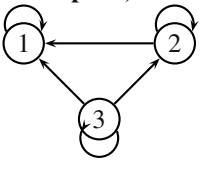
This relation between players can be seen as a directed graph containing a vertex for each player, and an edge from i to j whenever $j \in RP_i$, i.e. if j is a relevant player of i.

Definition 6. *Let* $G = (N, V, \pi, \Phi)$ *be a Boolean game. The* **dependency graph of a Boolean game** G *is the directed graph (but not necessariyly acyclic)* $\mathcal{P} = \langle N, R \rangle$*, with* $\forall i, j \in N$, $(i, j) \in R$ *(denoted by* $R(i, j)$*) if* $j \in RP_i$.

Thus, $R(i)$ is the set of players from which i may need some action in order to be satisfied: $j \in R(i)$ if and only if $j \in RP_i$. Remark however that $j \in R(i)$ does not imply that i *needs* some action by j to see her goal satisfied. For instance, if $\pi_1 = \{a\}$, $\pi_2 = \{b\}$ and $\varphi_1 = a \vee b$, then $1 \in R(2)$; however, 1 has a strategy for satisfying her goal (setting a to true) and therefore does not have to rely on 2.

We denote by R^* the transitive closure of R. $R^*(i, j)$ means that there exists a path from i to j in R. Then, $R^*(i)$ represents all players who have a direct or indirect influence on i. $R^{*-1}(i)$ represents all players on which i has a direct or indirect influence.

Example 1, continued: *The dependence graph* $\mathcal{P}$ *induced by* G *is depicted as follows:*



We have $R^{-1}(1) = \{1, 2, 3\}$, $R^{-1}(2) = \{2, 3\}$, $R^{-1}(3) = \{3\}$.
$R^*(1) = \{1\}$, $R^*(2) = \{1, 2\}$ *and* $R^*(3) = \{1, 2, 3\}$.
$R^{*-1}(1) = \{1, 2, 3\}$, $R^{*-1}(2) = \{2, 3\}$ *and* $R^{*-1}(3) = \{3\}$.

We easily obtain the following:

Proposition 1. *Every dependency graph represents at least one Boolean game.*

We now introduce the notion of stable set. A stable set is a set B of nodes such that all the edges from nodes in B get to another node in B. The set of relevant players of a stable set B are the players in B.

Definition 7. *Let* $G = (N, V, \pi, \Phi)$ *be a Boolean game.* $B \subseteq N$ *is* **stable** *for* R *if and only if* $R(B) \subseteq B$*, i.e.* $\forall j \in B$, $\forall i$ *such that* $i \in R(j)$*, then* $i \in B$.

Clearly, $\varnothing$ and N are stable, and the set of stable sets for a Boolean game is closed under union and intersection. These four properties actually fully characterize the set of coalitions that correspond to the set of stable coalitions for a Boolean game (recall that a coalition is a subset of N).

Proposition 2. *Let $\mathcal{C} \subset 2^N$. There exists a Boolean game G such that $\mathcal{C}$ is the set of stable sets for G if and only if $\mathcal{C}$ satisfies the following four properties: 1. $\varnothing \in \mathcal{C}$; 2. $N \in \mathcal{C}$; 3. If $B, B' \in \mathcal{C}$ then $B \cup B' \in \mathcal{C}$; 4. If $B, B' \in \mathcal{C}$ then $B \cap B' \in \mathcal{C}$.*

We now define the projection of a Boolean game G on the set of players $B \subseteq N$:

Definition 8. *Let $G = (N, V, \pi, \Phi)$ be a Boolean game, and $B \subseteq N$ a stable set for R. The* **projection** *of G on B is defined by $G_B = (B, V_B, \pi_B, \Phi_B)$, where $V_B = \cup_{i \in B} \pi_i$, $\pi_B : B \to V_B$ such that $\pi_B(i) = \{v | v \in \pi_i\}$, and $\Phi_B = \{\varphi_i | i \in B\}$.*

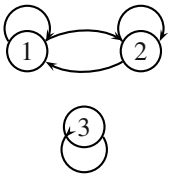
Proposition 3. *If B is a stable set, $G_B = (B, V_B, \pi_B, \Phi_B)$ is a Boolean game.*

Proof: Let $G_B = (B, V_B, \pi_B, \Phi_B)$. We have to check that every goal φ_i for $i \in B$ is a formula of L_{V_B}, or can be rewritten equivalently as a formula of L_{V_B}. Suppose than $\exists i \in B, \exists v \in Var(\varphi_i)$ such that $v \notin V_B$. So, $\forall j \in B$, $v \notin \pi_j$. Let $k \in N \setminus B$ such that $v \in \pi_k$. We know that $v \in Var(\varphi_i)$, so either φ_i is independent from v, and then is logically equivalent to a formula in which v does not appear; or φ_i is not independent from v, and in this case $v \in RV_i$ and by definition $k \in RP_i$. So, $k \in R(i)$, but $k \notin B$: this is in contradiction with the fact that B is stable. ■

Example 2. *Let $G = (N, V, \pi, \Phi)$ be the Boolean game defined by $V = \{a, b, c\}$, $N = \{1, 2, 3\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\varphi_1 = a \leftrightarrow b$, $\varphi_2 = a \leftrightarrow \neg b$ and $\varphi_3 = \neg c$.*

We have: $RV_1 = \{a, b\}$, $RV_2 = \{a, b\}$, $RV_3 = \{c\}$, $RP_1 = \{1, 2\}$, $RP_2 = \{1, 2\}$, $RP_3 = \{3\}$.

The dependency graph $\mathcal{P}$ of G follows. The sets of players $B = \{1, 2\}$ and $C = \{3\}$ are stable. We can decompose G in 2 Boolean games:



- $G_B = (B, V_B, \pi_B, \Phi_B)$, *with* $B = \{1, 2\}$, $V_B = \{a, b\}$, $\pi_1 = a$, $\pi_2 = b$, $\varphi_1 = a \leftrightarrow b$, $\varphi_2 = a \leftrightarrow \neg b$.
- $G_C = (C, V_C, \pi_C, \Phi_C)$, *with* $C = \{3\}$, $V_C = \{c\}$, $\pi_3 = c$, $\varphi_3 = \neg c$.

4 Nash Equilibria

Pure-strategy Nash equilibria (PNE) for n-players Boolean games are defined exactly as usual in game theory (see for instance [10]), having in mind that utility functions are induced from players' goals $\varphi_1, \ldots, \varphi_n$. A PNE is a strategy profile such that each player's strategy is an optimal response to the other players' strategies.

Definition 9. *Let $G = (N, V, \pi, \Phi)$ be a Boolean game with $N = \{1, \ldots, n\}$.*
$s = \{s_1, \ldots, s_n\}$ is a **pure-strategy Nash equilibrium** *(PNE) if and only if $\forall i \in \{1, \ldots, n\}$, $\forall s'_i \in S_i$, $u_i(s) \geq u_i(s_{-i}, s'_i)$.*

The following simple characterization of PNEs is straightforward from this definition ([4]): a strategy profile s is a pure-strategy Nash equilibrium for G iff for all $i \in N$, either $s \models \varphi_i$ or $s_{-i} \models \neg\varphi_i$ holds.

These definitions lead to some obvious properties of pure-strategy Nash equilibria:

Proposition 4. *Let G be a Boolean game. If $\forall i \in N$, $i \notin RP_i$ then every $s \in S$ is a PNE.*

If the irreflexive part of the players' dependency graph $\mathcal{P}$ of a game G is acyclic, (i.e. if there is no cycle of length ≥ 2), then we can use a procedure inspired by the "forward sweep procedure" [11] to find the pure-strategy Nash equilibria. Let us see this on an example.

Example 1, continued: *The irreflexive part of the dependency graph $\mathcal{P}$ of G is acyclic. $RP_1 = \{1\}$, so a strategy profile $s = (s_1, s_2, s_3)$ is a PNE only if 1's goal is satisfied, i.e., $s_1 = \overline{a}$. Then, 2 can choose her strategy, because her goal depends only on her and on 1. Thus, s is a PNE only if $(s_1, s_2) \models (\varphi_2)_{s_1}$, i.e., $s_2 = \overline{b}$. Finally, 3 knows the strategies of 1 and 2, and therefore she knows her goal will never be satisfied whatever she does. Therefore, G has 2 PNEs: $\{\overline{a}\overline{b}c, \overline{a}\overline{b}\overline{c}\}$.*

Proposition 5. *Let G be a Boolean game such that the irreflexive part of the dependency graph $\mathcal{P}$ of G is acyclic. Then, G has a PNE. Moreover, s is a PNE of G if and only if for every $i \in N$, either $(s_{R^*(i)\setminus\{i\}}, s_i) \models \varphi_i$ or $s_{R^*(i)\setminus\{i\}} \models \neg\varphi_i$.*

Obviously, when the irreflexive part of the dependency graph is not acyclic, the existence of PNE is no longer guaranteed (still, a game with a cyclic dependency graph may have a PNE, as shown in Example 3).

Proposition 5 leads to the following corollary:

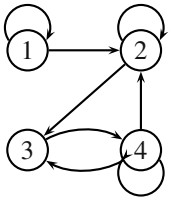
Corollary 1. *If G is a Boolean game such that $\forall i \in N$, $RP_i = \{i\}$, then s is a PNE if and only if $\forall i$, $s \models \varphi_i$.*

If G is a Boolean game such that $\forall i \in N$, $\exists j \in N$ such that $RP_i = \{j\}$, then s is a PNE if and only if $s \models \varphi_j$.

Proposition 6. *Let $G = (N, V, \pi, \Phi)$ be a Boolean game, $B \subseteq N$ a stable set for R, and G_B the projection of G on B. If s is a PNE for G, then s_B is a PNE for G_B.*

Example 3. *Let $G = (N, V, \pi, \Phi)$ be the Boolean game defined by $V = \{a, b, c, d\}$, $N = \{1, 2, 3, 4\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\pi_4 = \{d\}$, $\varphi_1 = a \leftrightarrow b$, $\varphi_2 = b \leftrightarrow c$, $\varphi_3 = \neg d$, and $\varphi_4 = d \leftrightarrow (b \wedge c)$. We have: $RP_1 = \{1, 2\}$, $RP_2 = \{2, 3\}$, $RP_3 = \{4\}$, $RP_4 = \{2, 3, 4\}$.*

The dependency graph $\mathcal{P}$ of G is the following:



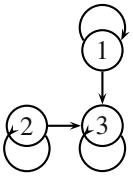
The set of players $B = \{2, 3, 4\}$ is stable. $G_B = (B, V_B, \pi_B, \Phi_B)$ is a Boolean game, with $V_B = \{b, c, d\}$, $\pi_2 = b$, $\pi_3 = c$, $\pi_4 = d$, $\varphi_2 = b \leftrightarrow c$, $\varphi_3 = \neg d$, and $\varphi_4 = d \leftrightarrow (b \wedge c)$.
G has 2 PNEs : $\{abcd, \overline{a}\overline{b}\overline{c}\overline{d}\}$, and $\{bcd, \overline{b}\overline{c}\overline{d}\}$ are 2 PNEs of G_B (and in this case, G_B has no other PNEs).

As we can see on Example 2, the converse is not always true: $C = \{3\}$ is stable, and the Boolean game $G_C = (C, V_C, \pi_C, \Phi_C)$ has a PNE : $\{\overline{c}\}$, but the game G has no PNE.

However, there exist simple cases for which the converse is true.

Proposition 7. *Let B and B' be two stable sets of players, and let G_B and $G_{B'}$ be the two Boolean games associated. Suppose than s_B is a PNE for G_B and $s_{B'}$ is a PNE for $G_{B'}$ such that $\forall i \in B \cap B'$, $s_{B,i} = s_{B',i}$, where $s_{B,i}$ represents the strategy of player i for the game G_B. Then, $s_{B\cup B'}$ is a PNE for $G_{B\cup B'}$.*

Example 4. *Let $G = (N, V, \pi, \Phi)$ be the Boolean game defined by $V = \{a, b, c\}$, $N = \{1, 2, 3\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\varphi_1 = a \leftrightarrow c$, $\varphi_2 = b \leftrightarrow \neg c$, and $\varphi_3 = c$. We have: $RP_1 = \{1, 3\}$, $RP_2 = \{2, 3\}$, $RP_3 = \{3\}$. The dependency graph $\mathcal{P}$ of G is drawn below. The sets of players $B = \{1, 3\}$ and $C = \{2, 3\}$ are stable. We have two new Boolean games.*



- *$G_B = (B, V_B, \pi_B, \Phi_B)$, with $B = \{1, 3\}$, $V_B = \{a, c\}$, $\pi_1 = a$, $\pi_3 = c$, $\varphi_1 = a \leftrightarrow c$ and $\varphi_3 = c$. G_B has one PNE : $\{ac\}$ (denoted by $s_B = (s_{B,1}, s_{B,3})$).*
- *$G_C = (C, V_C, \pi_C, \Phi_C)$, with $C = \{2, 3\}$, $V_C = \{b, c\}$, $\pi_2 = b$, $\pi_3 = c$, $\varphi_2 = b \leftrightarrow \neg c$, $\varphi_3 = c$. G_C has one PNE : $\{\overline{b}c\}$ (denoted by $s_C = (s_{C,2}, s_{C,3})$).*

$B \cap C = \{3\}$. But we have $s_{B,3} = s_{C,3} = c$: $G_{B\cup C}$ has one PNE: $\{a\overline{b}c\}$.

We can easily generalize Proposition 7, with p stable sets covering the set of players:

Proposition 8. *Let $G = (N, V, \pi, \Phi)$ be a Boolean game, and let $B_1 \ldots B_p$ be p stable sets of players, such that $B_1 \cup \ldots \cup B_p = N$. Let $G_{B_1}, \ldots, G_{B_p}$ be the p Boolean games associated. If $\exists s_{B_1} \ldots s_{B_p}$ PNEs of $G_{B_1}, \ldots, G_{B_p}$ such that $\forall i, j \in \{1, \ldots p\}$, $\forall k \in B_i \cap B_j$, $s_{B_i,k} = s_{B_j,k}$, then $s = (s_{B_1}, \ldots, s_{B_p})$ is a PNE of G.*

As shown in Example 4, splitting a Boolean game makes the computation of Nash equilibria easier. If we try to compute Nash equilibria in the original game, we have to check if either $s \models \varphi_i$ or $s_{-i} \models \neg\varphi_i$ for each of the 8 strategy profiles s and for each of the 3 players i. So, we have to make 12 verifications for each player (8 for each strategy profile in order to verify $s \models \varphi_i$, and 4 for each s_{-i} to verify $s_{-i} \models \neg\varphi_i$), then 36 for the game in the worst case. Meanwhile, the computation of PNEs once the game is split is much easier: for G_B, from Proposition 5, we have to make 6 verifications for player 1 (4 to compute $(s_1, s_3) \models \varphi_1$, and 2 to compute $s_3 \models \neg\varphi_1$); and only 2 for player 3 (because $R^*(3) \setminus \{3\} = \varnothing$). So, we only have to do 8 verifications in the worst case to find the PNEs of G_B, and the same for G_C, which has an equivalent configuration. As we have to check if the instanciation of player 3's variables are the same for PNEs of the 2 games, we have to make 17 verifications to compute PNEs of the game G.

5 Generalization to Non-dichotomous Preferences

This choice of binary utilities (where agents can express only plain satisfaction or plain dissatisfaction, with no intermediate levels) is a loss of generality. We would like now to allow for associating an arbitrary preference relation on S with each player. A *preference relation* $\succeq$ is a reflexive, transitive and complete binary relation on S. The strict preference $\succ$ associated with $\succeq$ is defined as usual by $s_1 \succ s_2$ if and only if $s_1 \succeq s_2$ and not $(s_2 \succeq s_1)$. Given a propositional language L for compact preference representation,

a L-Boolean game is defined a 4-uple $G = (N, V, \pi, \Phi)$, where $N = \{1, \ldots, n\}$, V, and π are as before and $\Phi = \langle \Phi_1, \ldots, \Phi_n \rangle$, where for each i, Φ_i is a compact representation, in L, of the preference relation $\succeq_i$ of agent i on S. We let $Pref_G = \langle \succeq_1, \ldots, \succeq_n \rangle$. Remark that if L_P is the purely propositional preference representation language, where a (dichotomous) preference is represented by a propositional formula, then L_P-Boolean games are just standard Boolean games as defined in Section 2. See [5] for several families of L-Boolean games.

We now have to generalize the dependency graph between players from Boolean games to L-Boolean games, for an arbitrary language L. Recall that, in Section 3, a player i was dependent on a player j if her propositional goal φ_i was dependent of one of the variables that j controls. Therefore, what we have to start with is generalizing formula-variable dependence to a dependency notion between a preference relation (or a syntactical input in a compact representation language from which this preference relation can be induced) and a variable. Several definitions have been considered in [12], in the more general case where preference relations are partial preorders. In the specific case where preference relations are complete preorders, however, there seems to be only one suitable definition: a preference relation $\succeq$ depends on a propositional variable x if there exists at least one state where the agent is not indifferent between this state and the state obtained by switching the value of x:

Definition 10. *A preference relation $\succeq$ on 2^V depends on a propositional variable $x \in V$ if there exists a $s \in S$ such that $switch(s,x) \succ_i s$ or $switch(s,x) \prec_i s$.*

This definition extends naturally to inputs of preference representation languages: an input Φ of a preference representation language L depends on x if the preference relation $\succeq$ induced by Φ depends on x.

We are now in position of defining the dependency graph for a L-Boolean game:

Definition 11. *Let $G = (N, V, \pi, \Phi)$ a L-Boolean game. The set of* **relevant variables** *for a player i, denoted by RV_i, is the set of all variables $v \in V$ such that Φ_i is dependent on v. The set of* **relevant players** *for a player i, denoted by RP_i, is the set of agents $j \in N$ such that j controls at least one relevant variable of i: $RP_i = \bigcup_{v \in RV_i} \pi^{-1}(v)$*

The dependency graph of a L-Boolean game is defined exactly as in Section 3.

These definitions do not depend on the language chosen for compact preference representation. However, for the sake of illustration we give an example in which preferences are represented with prioritized goals (see [5]):

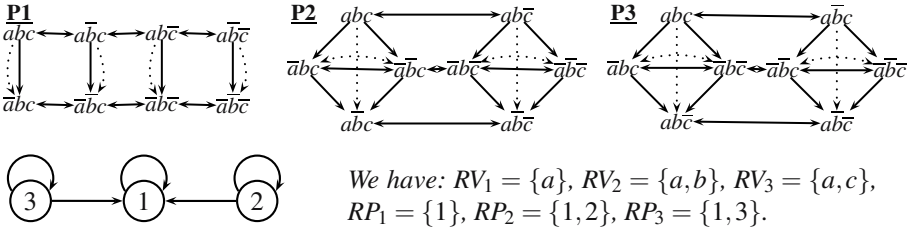
Definition 12. *A* **prioritized goal base** Σ *is a collection $\langle \Sigma^1; \ldots; \Sigma^p \rangle$ of sets of propositional formulas. Σ^j represents the set of goals of priority j, with the convention that the smaller j, the more prioritary the formulas in Σ^j.*

In this context, several criteria can be used in order to generate a preference relation $\succeq$ from Σ. We choose here to stick to the leximin criterion (see [13,14,15]). In the following, if s is an interpretation of 2^V then we let $Sat(s, \Sigma^j) = \{\varphi \in \Sigma^j \mid s \models \varphi\}$.

Definition 13. *Let $\Sigma = \langle \Sigma^1; \ldots; \Sigma^p \rangle$, and let s and s' be two interpretations of 2^V. The leximin preference relation is defined by: $s \succ^{lex} s'$ iff $\exists k \in \{1, \ldots, p\}$ such that: $|Sat(s, \Sigma^k)| > |Sat(s', \Sigma^k)|$ and $\forall j < k$, $|Sat(s, \Sigma^j)| = |Sat(s', \Sigma^j)|$.*

Note that $\succeq^{lex}$ is a complete preference relation. Here is now an example within this preference representation language:

Example 5. *$G = (N,V,\pi,\Phi)$ where $N = \{1,2,3\}$, $V = \{a,b,c\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\Sigma_1 = \langle a\rangle$, $\Sigma_2 = \langle (b \vee \neg a); a\rangle$ and $\Sigma_3 = \langle (c \vee \neg a); a\rangle$. We draw below the preference relations $Pref_G^{lex} = \langle \succeq_1^{lex}, \succeq_2^{lex}, \succeq_3^{lex}\rangle$[6].*



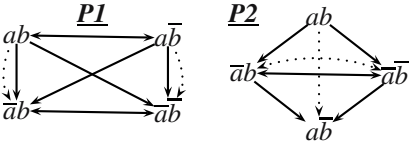
We have: $RV_1 = \{a\}$, $RV_2 = \{a,b\}$, $RV_3 = \{a,c\}$, $RP_1 = \{1\}$, $RP_2 = \{1,2\}$, $RP_3 = \{1,3\}$.

Definition 14. *Let $G = (N,V,\pi,\Phi)$ be a L-Boolean game, and $B \subseteq N$ a stable set for R. The* **projection** *of G on B is defined by $G_B = (B,V_B,\pi_B,\Phi_B)$, where $V_B = \cup_{i\in B}\pi_i$, $\pi_B(i) = \{v | v \in \pi_i\}$, and Φ_B are the goals of players in B.*

We can now generalize some properties found previously to these non-dichotomous preferences. For example, Propositions 3, 5, 6, 7 and 8 can be easily generalized in this framework.

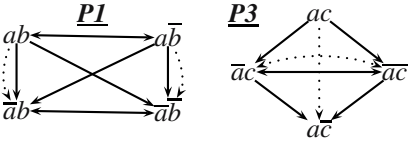
Example 5, continued: *The sets of players $B = \{1,2\}$ and $C = \{1,3\}$ are stable. We have two new Boolean games:*

$G_B = (B,V_B,\pi_B,\Phi_B)$, with $B = \{1,2\}$, $V_B = \{a,b\}$, $\pi_1 = a$, $\pi_2 = b$, $\Sigma_1 = \langle a\rangle$, and $\Sigma_2 = \langle (b \vee \neg a); a\rangle$. The preference relations $Pref_G^{lex} = \langle \succeq_1^{lex}, \succeq_2^{lex}\rangle$ are drawn on the right.



G_B has one PNE : $\{ab\}$ (denoted by $s_B = (s_{B,1}, s_{B,2})$).

$G_C = (C,V_C,\pi_C,\Phi_C)$, with $C = \{1,3\}$, $V_C = \{a,c\}$, $\pi_1 = a$, $\pi_3 = c$, $\Sigma_1 = \langle a\rangle$ and $\Sigma_3 = \langle (c \vee \neg a); a\rangle$. The preference relations $Pref_G^{lex} = \langle \succeq_1^{lex}, \succeq_3^{lex}\rangle$ are drawn on the right.



G_C has one PNE : $\{ac\}$ (denoted by $s_C = (s_{C,1}, s_{C,3})$).

$B \cap C = \{1\}$. But we have $s_{B,1} = s_{C,1} = a$: $G_{B\cup C}$ has one PNE: $\{abc\}$.

[6] Arrows are oriented from more preferred to less preferred strategy profiles (s_1 is preferred to s_2 is denoted by $s_1 \to s_2$). To make the figures clearer, we do not draw edges that are obtained from others by transitivity. The dotted arrows indicate the links taken into account in order to compute Nash equilibria. For example, player 2 prefers abc to $\overline{a}bc$ because $|Sat(ab,\Sigma_2^1)| = 1$, $|Sat(ab,\Sigma_2^2)| = 1$ (both stratas of Σ^2 are satisfied), and $|Sat(\overline{a}b,\Sigma_2^1)| = 1$, $|Sat(ab,\Sigma_2^2)| = 0$ (only the first strata of Σ_2 is satisfied).

6 Conclusion

We have shown how the intuitive notion of dependency between players in a Boolean game can be exploited so as to give simpler characterizations of pure-strategy Nash equilibria. Moreover, our properties not only hold for the standard version of Boolean game (with propositional goals and dichotomous preferences) but also for generalized Boolean games, where players' preferences are expressed in a compact representation language. Another class of games with dichotomous preferences shares a lot with Boolean games: Qualitative Coalitional Games (QCG), introduced by [16]. In a QCG, each agent has a set of goals, and is satisfied if one of her goals is achieved, but is indifferent on which goal is, and on the number of goals achieved[7]. Thus agents have dichotomous preferences (as in the standard version of Boolean games - cf. Sections 2–4). A characteristic function associates with each agent, or set of agents, the set of goals they can achieve. The main difference between QCGs and BG is that characteristic functions in QCGs are not monotonic, whereas utility functions are in Boolean games. However, we can represent a QCG with monotonic characteristic function by a Boolean games.

Boolean games take place in a larger stream of work, that we may gather under the generic name of *compactly represented games*. All frameworks for compactly represented games make use of notions of dependencies between players and/or actions that have a lot in common with ours. Most of these frameworks, including [6,7,18], share the following mode of representation of players' utilities: the utility of a player i is described by a table specifying a numerical value for each combination of values to each of the set of variables that are relevant to i[8]. The representation of games with such utility tables is called *graphical normal form* (GNF) in [8]. Dependency between players and variables in such games naturally induce a dependency relation between players, in the same way as we do (i depends on j if i's utility table refers to a variable that is controlled by j).

Boolean games are very similar to these graphical games, except that the form chosen for expressing compactly players' preferences is *logical*. The logical form is sometimes exponentialy more compact than the graphical form: consider for instance the dichotomous preference relation corresponding to the goal $\varphi = x_1 \oplus \ldots \oplus x_p$, where $\oplus$ is exclusive or. While the logical representation of u_φ is linear in p, its representation by utility tables would be exponential in p, since each of the p variables is relevant to the utility of the player. In the general case of non-dichotomous utility functions or preference relations, the Boolean game framework, by allowing some flexibility on the choice of the language for preference representation, is more general than that of graphical games, where the format for expressing preferences is fixed. Moreover, solving games in logical form may benefit from the huge literature on SAT and related algorithms for propositional logic.

The notion of dependency between players and variables in graphical games is used for the very same purpose as our dependency graph, namely, to split up a game into a set of interacting smaller games, which can solved more or less independently. [8] study speficic restrictions on graphical games, either by bounding the size of players' neigh-

[7] In [17], QCGs are extended by allowing agents to have preferences over goals.

[8] In multi-agent influence diagrams [6], a players' utility is actually express in a more compact way as the sum of local utilities, each corresponding to a smaller set of variables.

bourhoods (the neighbourhood of a player i in a graphical game is the set of players who potentially influence the utility of i), or by imposing that the dependency relation between players should be acyclic. They study the impact of such restrictions on the complexity of checking the existence of a Nash equilibrium (or their computation). Clearly, similar structural restrictions on Boolean games would probably allow for a complexity fall with respect to the complexity results for the general case in [4]. This is left for further study.

The work reported here is still preliminary and can be pursued in many other directions.

First, apart of the *structural* restrictions mentioned just above, we may study the impact of *syntactical* restrictions on propositional goals on the computation of Nash equilibria and on the construction of the dependency graph. In [19], Sichman and Conte introduced dependence graphs which can represent and/or dependencies[9] on actions needed to achieve an agent's goal and on the agents who control these actions. In the first case, this is similar to our set of relevant variables, and in the second case this corresponds to our set of relevant players. Sichman and Conte's ideas can be used for introducing and/or dependencies in our framework, but using the syntactical form of the goals. In [20], 3 notions of dependance are defined: the weak one is the same than our (an agent i is dependent from a set of agents C if C can achieve i's goal). The second one, normal dependence, adds to weak dependence the condition that i cannot achieve her goal by herself. Finally, the third one adds the fact that agents in C are the only ones able to achive i's goal. Following [19], [20] use an and-graph to reprensent weak/strong dependence: for every coalition C, there is an and-edge from agent i, $i \in C$, to agent $j \in N$ if the agents in C can achieve the goal desired by the agent j. This notion of dependence is the basis of their computation of admissible coalition under the do-ut-des criterion (see [21]).

Second, while our Section 5 does not focus on particular language (prioritized goals we used in an example just for the sake of illustration), we may want to study in further detail the computation of Nash equilibria (using the structural properties of the game) for some specific languages for preference representation (see [5] for the case of CP-nets and prioritized goals). A particularly appealing language is that of *weighted goals*, where a player's utility function is represented using several propositional formulas, each being attached with a numerical value (see [22]). This is especially interesting because this language generalizes the representation by utility tables in graphical games.

So far, Boolean games allow only for expressing *static* games (with simultaneous moves by the players) and with *complete information*. Enriching Boolean games with dynamicity and nature-driven uncertainty, as in multi-agent influence diagrams, is not as simple as it looks at first glance, and is a challenging issue. Computing *mixed strategy* Nash equilibria in Boolean games is another challenging issue.

Acknowledgment

We thank the anonymous reviewers for their useful comments and suggestions.

[9] The or-dependence means that several actions allow an agent to achieve a in several ways, and the and-dependence means that this agent needs all these actions to achieve her goal.

References

1. Harrenstein, P., van der Hoek, W., Meyer, J., Witteveen, C.: Boolean Games. In: Proc. of TARK'01, pp. 287–298. Morgan Kaufmann, San Francisco (2001)
2. Harrenstein, P.: Logic in Conflict. PhD thesis, Utrecht University (2004)
3. Dunne, P., van der Hoek, W.: Representation and Complexity in Boolean Games. In: Alferes, J.J., Leite, J.A. (eds.) JELIA 2004. LNCS (LNAI), vol. 3229, pp. 347–359. Springer, Heidelberg (2004)
4. Bonzon, E., Lagasquie-Schiex, M., Lang, J., Zanuttini, B.: Boolean games revisited. In: Proc. of ECAI'06, pp. 265–269. Springer, Heidelberg (2006)
5. Bonzon, E., Lagasquie-Schiex, M., Lang, J.: Compact preference representation for Boolean games. In: Yang, Q., Webb, G. (eds.) PRICAI 2006. LNCS (LNAI), vol. 4099, pp. 41–50. Springer, Heidelberg (2006)
6. Koller, D., Milch, B.: Multi-Agent Influence Diagrams for Representing and Solving Games. Games and Economic Behavior 45(1), 181–221 (2003)
7. Kearns, M., Littman, M.L., Singh, S.: Graphical Models for Game Theory. In: Procs. of UAI'01 (2001)
8. Gottlob, G., Greco, G., Scarcello, F.: Pure Nash Equilibria: Hard and Easy Games. Journal of Artificial Intelligence Research 24, 357–406 (2005)
9. Lang, J., Liberatore, P., Marquis, P.: Propositional Independence - Formula-Variable Independence and Forgetting. Journal of Artificial Intelligence Research 18, 391–443 (2003)
10. Osborne, M., Rubinstein, A.: A course in game theory. MIT Press, Cambridge (1994)
11. Boutilier, C., Brafman, R., Hoos, H., Poole, D.: Reasoning with Conditional Ceteris Paribus Preference Statements. In: Proc. of UAI'99 (1999)
12. Besnard, P., Lang, J., Marquis, P.: Variable Forgetting in Preference Relations over Propositional Domains. In: Proc. of ECAI'06, pp. 763–764. Springer, Heidelberg (2006)
13. Dubois, D., Lang, J., Prade, H.: Inconsistency in possibilistic knowledge bases: To live with it or not live with it. In: Fuzzy Logic for the Management of Uncertainty, pp. 335–351 (1992)
14. Benferhat, S., Cayrol, C., Dubois, D., Lang, J., Prade, H.: Inconsistency management and prioritized syntax-based entailment. In: Proc. of IJCAI'93, pp. 640–645 (1993)
15. Lehmann, D.: Another Perspective on Default Reasoning. Annals of Mathematics and Artificial Intelligence 15, 61–82 (1995)
16. Wooldridge, M., Dunne, P.: On the computational complexity of qualitative coalitional games. Artificial Intelligence (2004)
17. Dunne, P., Wooldridge, M.: Preferences in Qualitative Coalitional Games. In: Proc. of GTDT'04, pp. 29–38 (2004)
18. Mura, P.L.: Game Networks. In: Procs. of UAI'00, pp. 335–342 (2000)
19. Sichman, J., Conte, R.: Multi-Agent Dependence by Dependence Graphs. In: Alonso, E., Kudenko, D., Kazakov, D. (eds.) Adaptive Agents and Multi-Agent Systems. LNCS (LNAI), vol. 2636, Springer, Heidelberg (2003)
20. Boella, G., Sauro, L., van der Torre, L.: From Social Power to Social Importance. Web Intelligence and Agent Systems Journal (to appear)
21. Boella, G., Sauro, L., van der Torre, L.: Admissible agreements among goal-directed agents. In: IEEE (ed.) Proc. of IAT'05 (2005)
22. Chevaleyre, Y., Endriss, U., Lang, J.: Expressive Power of Weighted Propositional Formulas for Cardinal Preference Modelling. In: Proc. of KR'06, pp. 145–152. AAAI Press, Stanford (2006)

The Use of Fuzzy t-Conorm Integral for Combining Classifiers*

David Štefka and Martin Holeňa

Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic
{stefka,martin}@cs.cas.cz

Abstract. Choquet or Sugeno fuzzy integrals are commonly used for aggregating the results of different classifiers. However, both these integrals belong to a more general class of fuzzy t-conorm integrals. In this paper, we describe a framework of a fuzzy t-conorm integral and its use for combining classifiers. We show the advantages of this approach to classifier combining in several benchmark tests.

Keywords: combining classifiers, classifier aggregation, fuzzy integral, t-conorm integral, fuzzy measure.

1 Introduction

Combining several different classifiers in order to improve the quality of classification is a common approach today and many different methods have been described in the literature. Out of these, fuzzy integral is often used. Although there is a general class of fuzzy t-conorm integrals, introduced by Murofushi and Sugeno [1], only two particular integrals (Choquet integral and Sugeno integral) from this class are commonly used.

In this paper, we describe a framework of fuzzy t-conorm integral, investigate which particular types of fuzzy t-conorm systems are useful for combining classifiers, and perform tests on real data to show the performance of combining classifiers by using fuzzy t-conorm integral.

The paper is structurred as follows: in Section 2, we introduce the formalism needed for combining classifiers. Section 3 deals with fuzzy integrals and fuzzy measures. Subsection 3.1 introduces the fuzzy t-conorm integral, Subsection 3.2 describes the way a fuzzy t-conorm integral can be used for combining classifiers, and Subsection 3.3 investigates different t-conorm systems for integration and considers which particular systems are useful for classifier combining. Section 4 contains experimental results, and finally, Section 5 then concludes the paper.

* The research reported in this paper was partially supported by the Program "Information Society" under project 1ET100300517 (D. Štefka) and the grant No. 201/05/0325 of the Grant Agency of the Czech Republic (M. Holeňa), and partially supported by the Institutional Research Plan AV0Z10300504.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 755–766, 2007.

2 Combining Classifiers

We define a classifier as a mapping $\phi : \mathcal{X} \to [0,1]^N$, where $\mathcal{X} \subseteq \mathbb{R}^n$ is a n-dimensional feature space, and $\phi(\boldsymbol{x}) = [\mu_1(\boldsymbol{x}), \ldots, \mu_N(\boldsymbol{x})]$ is a vector of "weights of classification" of the pattern $\boldsymbol{x}$ to each class $C_1, \ldots, C_N$. This type of classifier is called *measurement classifier* [2] or *possibilistic classifier* [3] in the literature.

Whenever we want to use classifier combining to improve the quality of the classification, first of all, we need to create a team of diverse classifiers $\phi_1, \ldots, \phi_k$. Most often, the team consists of classifiers of the same type, which differ only in their parameters, dimensionality, or training sets – such a team is usually called an *ensemble* of classifiers. The restriction to classifiers of the same type is not essential, but it ensures that the classifiers' outputs are consistent. Well-known methods for ensemble creation are *bagging* [4], *boosting* [5], or *multiple feature subset* (MFS) methods [6].

After we have constructed an ensemble of classifiers $\phi_1, \ldots, \phi_k$, we have to use some function $\mathcal{A}$ to aggregate the results of the individual classifiers to get the final prediction, i.e. $\Phi(\mathbf{x}) = \mathcal{A}(\phi_1(\mathbf{x}), \ldots, \phi_k(\boldsymbol{x}))$, where Φ is the final aggregated classifier. The output of an ensemble can be structured to a $k \times N$ matrix, called *decision profile* (DP):

$$DP(\boldsymbol{x}) = \begin{pmatrix} \phi_1(\boldsymbol{x}) \\ \phi_2(\boldsymbol{x}) \\ \vdots \\ \phi_k(\boldsymbol{x}) \end{pmatrix} = \begin{pmatrix} \mu_{1,1}(\boldsymbol{x}) & \mu_{1,2}(\boldsymbol{x}) & \ldots & \mu_{1,N}(\boldsymbol{x}) \\ \mu_{2,1}(\boldsymbol{x}) & \mu_{2,2}(\boldsymbol{x}) & \ldots & \mu_{2,N}(\boldsymbol{x}) \\ & & \ddots & \\ \mu_{k,1}(\boldsymbol{x}) & \mu_{k,2}(\boldsymbol{x}) & \ldots & \mu_{k,N}(\boldsymbol{x}) \end{pmatrix}, \tag{1}$$

The $i-$th row of the $DP(\boldsymbol{x})$ is an output of the corresponding classifier ϕ_i, and the $j-$th column contains the weights of classification of $\boldsymbol{x}$ to the corresponding class C_j given by all the classifiers.

Many methods for aggregating the ensemble of classifiers into one final classifier have been reported in the literature. A good overview of the commonly used aggregation methods can be found in [3]. These methods comprise simple arithmetic rules (voting, sum, product, maximum, minimum, average, weighted average, cf. [3,7]), fuzzy integral [3,8], Dempster-Shafer fusion [3,9], second-level classifiers [3], decision templates [3], and many others.

In this paper, we deal with the fuzzy integral for classifier combining. In the literature, the Sugeno or Choquet integral are commonly used for classifier aggregation, but there is also a more general framework for fuzzy integral by Murofushi and Sugeno [1], which contains Sugeno and Choquet integral as a special case. This framework is called *fuzzy t-conorm integral*, and is described in the next section.

3 Fuzzy Integral and Fuzzy Measure

Through the rest of the paper, we use the following notation for common t-norms and t-conorms:

- Standard: $x \wedge_S y = \min(x, y), x \vee_S y = \max(x, y)$
- Łukasiewicz: $x \wedge_L y = \max(x + y - 1, 0), x \vee_L y = \min(x + y, 1)$
- Product: $x \wedge_P y = xy, x \vee_P y = x + y - xy$
- Drastic: $x \wedge_D y = x$ if $y = 1$, y if $x = 1$, and 0 otherwise; $x \vee_D y = x$ if $y = 0$, y if $x = 0$, and 1 otherwise.

In general, fuzzy integrals (see [10] for details) can be looked upon as aggregation operators with respect to a fuzzy measure. The integrand (function values to aggregate) is integrated with the the fuzzy measure (expressing the importance of individual elements). A fuzzy measure is a generalization of the classical probability measure (the difference is that it does not need to fulfil the conditions of σ-additivity), and can be defined as follows:

Definition 1. *Let X be a nonempty set, Ω a set of subsets of X, such that $\emptyset, X \in \Omega$. A* fuzzy measure *over (X, Ω) is a function $g : \Omega \to [0, 1]$, such that:*

1. $g(\emptyset) = 0, g(X) = 1$, *and*
2. *if $A, B \in \Omega, A \subseteq B$, then $g(A) \leq g(B)$.*

The tuple (X, Ω, g) is called a fuzzy measure space.

If $X = \{x_1, \ldots, x_k\}$ is a finite set with k elements and Ω is the power set (set of all subsets) of X, then the fuzzy measure is determined by its 2^k values. Sugeno introduced the so-called λ-*fuzzy measure* [11], which needs only k values $g(x_1) = g_1, \ldots, g(x_k) = g_k$ to be determined properly (these values are called *fuzzy densities*), and the remaining values are computed using

$$g(A \cup B) = g(A) + g(B) + \lambda g(A)g(B), \tag{2}$$

where $A, B \in \Omega, A \cap B = \emptyset$, and λ is the only non-zero (if the fuzzy densities do not sum to one; if they do, $\lambda = 0$) root greater than -1 of the equation

$$\lambda + 1 = \prod_{i=1}^{k} (1 + \lambda g_i). \tag{3}$$

After we have defined a fuzzy measure, we can define the two commonly used fuzzy integrals. These are Choquet fuzzy integral and Sugeno fuzzy integral. Since for classifier combining X is a finite set, we restrict ourselves to so-called *simple functions*:

Definition 2. *A function $f : X \to [0, 1]$ is* simple, *if there exist $n \in \mathbb{N}$, $a_1, \ldots, a_n \in [0, 1], a_1 \leq a_2 \leq \cdots \leq a_n$, and $D_1, \ldots, D_n \subseteq X, D_i \cap D_j = \emptyset$ for $i \neq j$, such that $\forall x \in X$:*

$$f(x) = \sum_{i=1}^{n} a_i \mathbf{1}_{D_i}(x), \tag{4}$$

where $\mathbf{1}_{D_i}$ is the characteristic function of D_i.

Equivalent form of (4) is:

$$f(x) = \sum_{i=1}^{n} (a_i - a_{i-1}) \mathbf{1}_{A_i}(x), \tag{5}$$

where $a_0 = 0$ *and* $A_i = \bigcup_{j=i}^{n} D_j$.

Definition 3. *Let* g *be a fuzzy measure from Def. 1,* f *a simple function with* a_i *and* $A_i \in \Omega, i = 1, \ldots n$ *from Def. 2. Then the* Sugeno integral *of* f *with respect to a fuzzy measure* g*, denoted as* $\int_S f \mathrm{d}g$*, is defined by:*

$$\int_S f \mathrm{d}g = \bigvee_{S\ i=1}^{n} (a_i \wedge_S g(A_i)). \tag{6}$$

Definition 4. *Let* g *be a fuzzy measure from Def. 1,* f *be a simple function with* a_i *and* $A_i \in \Omega, i = 1, \ldots n$ *from Def. 2. Then the* Choquet integral *of* f *with respect to a fuzzy measure* g*, denoted as* $\int_C f \mathrm{d}g$*, is defined by:*

$$\int_C f \mathrm{d}g = \sum_{i=1}^{n} (a_i - a_{i-1}) g(A_i). \tag{7}$$

3.1 The Fuzzy t-Conorm Integral

Although Sugeno and Choquet fuzzy integrals are used routinely in many applications, they belong to the class of the so-called *fuzzy t-conorm integrals*, which were introduced in [1]. In this section, we present the framework of t-conorm fuzzy integral for simple functions, following [10]. For details and further information about the following definitions, refer to [1,10].

The individual types of fuzzy t-conorm integrals differ in the way how they bind together the spaces of integrand, measure, and integral. To formalize this, the spaces are linked together by the following definition.

Definition 5. *Let* $\triangle, \perp, \underline{\perp}$ *be continuous t-conorms, each of which is either Archimedean, or* $\vee_S$*. Let* $([0,1], \triangle), ([0,1], \perp)$*, and* $([0,1], \underline{\perp})$ *denote the spaces of values of integrand, measure, and integral, respectively. Let* $\odot : ([0,1], \triangle) \times ([0,1], \perp) \to ([0,1], \underline{\perp})$ *be a non-decreasing operation in both variables satisfying the following:*

1. $\odot$ *is continuous on* $(0,1]^2$*,*
2. $a \odot x = 0$ *if and only if* $a = 0$ *or* $x = 0$*,*
3. *If* $x \perp y < 1$*, then* $\forall a \in [0,1] : a \odot (x \perp y) = (a \odot x) \underline{\perp} (a \odot y)$*,*
4. *If* $a \triangle b < 1$*, then* $\forall x \in [0,1] : (a \triangle b) \odot x = (a \odot x) \underline{\perp} (b \odot x)$*.*

Then $\mathcal{F} = (\triangle, \perp, \underline{\perp}, \odot)$ *is called a* t-conorm system for integration*. If all the three t-conorms* $\triangle, \perp, \underline{\perp}$ *are Archimedean,* $\mathcal{F}$ *is then called* Archimedean*.*

Prior to defining the t-conorm integral for simple functions, we have to define the *pseudo-difference*.

Definition 6. *Let $\triangle$ be a t-conorm. An operation $-_\triangle : [0,1]^2 \to [0,1]$, defined as*

$$a -_\triangle b = \inf\{c | b \triangle c \geq a\}, \tag{8}$$

is called pseudo-difference *of a and b with respect to $\triangle$.*

Remark 1. Pseudo-difference is dual to the residue $\Rightarrow_\wedge$, defined as $a \Rightarrow_\wedge b = \sup\{c | a \wedge c \leq b\}$, $\wedge$ being a t-norm.

Example 1. If $\triangle = \vee_S$, then

$$a -_{\vee_S} b = \begin{cases} a & \text{if } a \geq b \\ 0 & \text{if } a < b \end{cases} \tag{9}$$

Example 2. If $\triangle = \vee_L$, then

$$a -_{\vee_L} b = \max(a - b, 0) \tag{10}$$

Now we are ready to define fuzzy t-conorm integral.

Definition 7. *Let (X, Ω, g) be a fuzzy measure space, $\mathcal{F} = (\triangle, \perp, \underline{\perp}, \odot)$ a t-conorm system for integration and f a simple function with a_i and $A_i, i = 1, \ldots n$ from Def. 2. The* fuzzy t-conorm integral *of f based on $\mathcal{F}$ with respect to g is defined by:*

$$\int_{\mathcal{F}} f \odot \mathrm{d}g = \underline{\perp}_{i=1}^{n}((a_i -_\triangle a_{i-1}) \odot g(A_i)). \tag{11}$$

Example 3. For $\mathcal{F} = (\vee_S, \vee_S, \vee_S, \wedge_S)$, we get the Sugeno integral.

Example 4. For $\mathcal{F} = (\vee_L, \vee_L, \vee_L, \cdot)$, where $\cdot$ is the ordinary multiplication, we get the Choquet integral.

3.2 Using Fuzzy Integral for Classifier Aggregation

Suppose we have a team of classifiers $\phi_1, \ldots, \phi_k : \mathcal{X} \to [0,1]^N$ and a t-conorm system for integration $\mathcal{F} = (\triangle, \perp, \underline{\perp}, \odot)$. For a given pattern $\boldsymbol{x} \in \mathcal{X}$, we organize the outputs of the classifiers in the team to a decision profile (1) $DP(\boldsymbol{x})$, and for each class $C_j, j = 1, \ldots, N$, we fuzzy-integrate the j-th column of $DP(\boldsymbol{x})$, resulting in the aggregated weight of the classification of $\boldsymbol{x}$ to the class C_j.

In this case, the space (universe) X from Def. 1 and 2 is the set of all classifiers, i.e. $X = \{\phi_1, \ldots, \phi_k\}$, and the j-th column of $DP(\boldsymbol{x})$ is a simple function $f : X \to ([0,1], \triangle)$.

To obtain $a_i, D_i, A_i, i = 1, \ldots, k$ from Def. 2, we sort the values in j-th column of $DP(\boldsymbol{x})$ in ascending order, and we denote the sorted values $a_1, \ldots, a_k$. In other words, a_1 is the lowest weight of classification of $\boldsymbol{x}$ to C_j (acquired by some classifier, which we will denote $\phi_{(1)}$), a_2 the second lowest (acquired by $\phi_{(2)}$), and so on.

Using this notation, $f(\phi_{(i)}) = a_i$, and moreover

$$f(x) = \sum_{i=1}^{k} a_i \mathbf{1}_{D_i}(x) = \sum_{i=1}^{k} (a_i - a_{i-1}) \mathbf{1}_{A_i}(x), \tag{12}$$

where $x \in X = \{\phi_1, \ldots, \phi_k\}$, and $a_0 = 0 \leq a_1 \leq a_2 \leq \cdots \leq a_k$, $D_i = \{\phi_{(i)}\}$, $A_i = \bigcup_{l=i}^{k} D_l$, which corresponds to Def. 2.

So far, we have properly determined the function f to integrate (integrand), and the space of integrand values $([0,1], \triangle)$. Now we need to define a fuzzy measure g. Again, let $X = \{\phi_1, \ldots, \phi_k\}$ be the universe, and let Ω be the power set of X, and $g : \Omega \to ([0,1], \perp)$ a mapping satisfying Def. 1. In the rest of the paper, we use the λ-fuzzy measure, but any other fuzzy measure could be used.

As can be seen from (11), we do not need all the 2^k values of g for the integration – only $g(A_i), i = 1, \ldots, k$ are needed. Let g be a λ-fuzzy measure, defined in Section 3 (replacing g_i by the permuted $g_{(i)}$), where $g_{(i)} = g(\phi_{(i)}) \in [0,1], i = 1, \ldots, k$ represent the importance of individual classifiers. These values are called *fuzzy densities*, and can be defined for example as $g_{(i)} = 1 - Err(\phi_i)$, where $Err(\phi_i)$ denotes the error rate of classifier ϕ_i. To compute all the necessary values of g, we use the following recursive approach based on (2), for $l = k, \ldots, 3$:

$$\begin{aligned} g(A_k) &= g(D_k) = g(\{\phi_{(k)}\}) = g_{(k)} \\ g(A_{l-1}) &= g(D_{l-1} \cup A_l) = g(\{\phi_{(l-1)}, \ldots, \phi_{(k)}\}) = \\ &= g(A_l) + g_{(l-1)} + \lambda g(A_l) g_{(l-1)} \\ &\ldots \\ g(A_1) &= g(D_1 \cup A_2) = g(\{\phi_{(1)}, \ldots, \phi_{(k)}\}) = \\ &= g(A_2) + g_{(1)} + \lambda g(A_2) g_{(1)} \end{aligned} \tag{13}$$

Now, having properly identified the function to integrate f and the fuzzy measure g, we can finally compute the fuzzy t-conorm integral of f based on $\mathcal{F}$, with respect to g, according to (11). The result of the integration is the aggregated weight of the classification of $\boldsymbol{x}$ to the class C_j. The whole process is summarized in Fig. 1.

3.3 Classification of t-Conorm Systems

The framework of fuzzy t-conorm integral provides many different types of fuzzy integrals, depending on the t-conorm system for integration used. However, not all combinations of t-conorms give t-conorm systems for integration, and moreover, not all t-conorm systems for integration provide useful approach to combining classifiers. In this section, we classify the t-conorm systems into classes, and investigate each class in detail.

One important class of t-conorm systems are Archimedean systems, i.e. $\mathcal{F} = (\triangle, \perp, \underline{\perp}, \odot)$, where all the t-conorms are Archimedean. As noticed in [10], if $\mathcal{F}$ is Archimedean, the corresponding fuzzy integral can be expressed as Choquet integral (and hence it has nearly the same properties as the Choquet integral).

Input: Team of classifiers $\phi_1, \ldots, \phi_k$, t-conorm system for integration $\mathcal{F}$, fuzzy densities $g_1, \ldots, g_k$ (e.g. $g_i = 1 - Err(\phi_i)$), pattern $\boldsymbol{x}$ to classify.
Output: $\Phi(\boldsymbol{x}) = [\mu_1(\boldsymbol{x}), \ldots, \mu_N(\boldsymbol{x})]$, i.e. a vector of weights of classification of $\boldsymbol{x}$ to all the classes $C_1, \ldots, C_N$.

1. Let each of the classifiers $\phi_1, \ldots, \phi_k$ predict independently, resulting in a decision profile $DP(\boldsymbol{x})$, of the form (1).
2. If $\sum_{i=1}^{k} g_i = 1$, set $\lambda = 0$, otherwise calculate λ as the only non-zero root greater than -1 of the polynomial equation (2).
3. For $j = 1, \ldots, N$ aggregate the j-th column of $DP(\boldsymbol{x})$ (i.e. weights of classification of $\boldsymbol{x}$ to C_j from all the classifiers) using fuzzy t-conorm integral:
   (a) Sort the values in the j-th column of $DP(\boldsymbol{x})$ in ascending order, denoting the sorted values $a_1, \ldots, a_k$. The corresponding classifiers will be denoted $\phi_{(1)}, \ldots, \phi_{(k)}$, and the corresponding fuzzy densities $g_{(1)}, \ldots, g_{(k)}$.
   (b) Using (13), compute $g(A_i), i = 1, \ldots, k$, where $A_i = \{\phi_{(i)}, \ldots, \phi_{(k)}\}$.
   (c) Using (11), compute the aggregated weight of classification of $\boldsymbol{x}$ to class C_j:

$$\mu_j(\boldsymbol{x}) = \int_{\mathcal{F}} f \odot \mathrm{d}g = \underline{\perp}_{i=1}^{n}((a_i -_{\triangle} a_{i-1}) \odot g(A_i)). \tag{14}$$

4. End with output $\Phi(\boldsymbol{x}) = [\mu_1(\boldsymbol{x}), \ldots, \mu_N(\boldsymbol{x})]$.

Fig. 1. Aggregation of classifier team using fuzzy t-conorm integral with respect to a λ-fuzzy measure

Let $h_{\triangle}, h_{\perp}, h_{\underline{\perp}}$ denote the generators of $\triangle, \perp, \underline{\perp}$, respectively, then the following holds:

$$\int_{\mathcal{F}} f \odot \mathrm{d}g = h_{\underline{\perp}}^{-1}\left[\min\left(h_{\underline{\perp}}(1), \int_C h_{\triangle} \circ f \mathrm{d}(h_{\perp} \circ g)\right)\right], \tag{15}$$

where $\circ$ denotes function composition. Therefore, we will focus our attention to non-Archimedean t-conorm systems in the rest of the paper.

From the non-Archimedean t-conorm systems (i.e. at least one of the t-conorms is $\vee_S$, and the rest is Archimenean), we will set aside systems with $\triangle \neq \underline{\perp}$. The reason for this is that if integral is regarded as mean value of integrands, then the spaces of integrand $([0,1], \triangle)$ and integral $([0,1], \underline{\perp})$ must be the same, i.e. $\triangle = \underline{\perp}$ (see [1,10] for details). However, this class of t-conorm systems is not as large as it may seem, because of the following proposition.

Proposition 1. *Let $\mathcal{F} = (\triangle, \perp, \underline{\perp}, \odot)$ be a non-Archimedean t-conorm system for integration, such that $\triangle = \underline{\perp}$, and $\odot$ is not constant in the second argument on (0,1]. Then $\triangle = \perp = \underline{\perp} = \vee_S$.*

Proof. Since $\mathcal{F}$ is not Archimedean, i.e. at least one of $\triangle, \perp, \underline{\perp}$ is $\vee_S$ and the rest is Archimedean, and $\triangle = \underline{\perp}$, there are only two situations possible:

- $\mathcal{F} = (\triangle, \vee_S, \triangle, \odot)$, where $\triangle$ is $\vee_S$, or continuous Archimedean t-conorm. Suppose that $\triangle \neq \vee_S$, i.e. it is continuous and Archimedean. According to Req. 3 from Def. 5, when $x \vee_S y < 1$, then $\forall a \in [0,1]$

$$a \odot (x \vee_S y) = (a \odot x) \triangle (a \odot y). \tag{16}$$

Let $x = y < 1$. Then (16) reduces to

$$a \odot x = (a \odot x) \triangle (a \odot x). \tag{17}$$

This conflicts with the fact that $\triangle$ is Archimedean, i.e. $u \triangle u > u \; \forall u \in (0,1)$.

– $\mathcal{F} = (\vee_S, \perp, \vee_S, \odot)$, where $\perp$ is $\vee_S$, or continuous Archimedean t-conorm. Suppose that $\perp \neq \vee_S$, i.e. it is continuous and Archimedean. According to Req. 3 from Def. 5, when $x \perp y < 1$, then $\forall a \in [0,1]$

$$a \odot (x \perp y) = (a \odot x) \vee_S (a \odot y). \tag{18}$$

Let $x = y$, such that $x \perp x < 1$. Then (18) reduces to

$$a \odot (x \perp x) = a \odot x. \tag{19}$$

Since $\perp$ is Archimedean, this is with conflict with the fact that $\odot$ is not constant in the second argument on (0,1].

In both cases $\triangle = \perp = \underline{\perp} = \vee_S$, which proves the proposition. □

We could still construct t-conorm systems for integration of the form $\mathcal{F} = (\vee_S, \perp, \vee_S, \odot)$, with $\odot$ constant in the second argument (because then (19) holds), i.e., in fact, integrals with no respect to the measure. However, this is not very useful. Fuzzy t-conorm systems of the form $\mathcal{F} = (\vee_S, \vee_S, \vee_S, \odot)$ are called $\vee_S$*-type* systems. The following proposition expresses that for $\vee_S$-type systems, Req. 3 and 4 from Def. 5 are satisfied automatically.

Proposition 2. *Let* $\odot : [0,1] \times [0,1] \to [0,1]$ *be a non-decreasing operation satisfying requirements 1 and 2 from Def. 5. Then* $\mathcal{F} = (\vee_S, \vee_S, \vee_S, \odot)$ *is a t-conorm system for integration.*

Proof. We have to prove the Req. 3 and 4 from Def. 5. Since the proof of the latter is analogous to the proof of the former, we will prove only Req. 3, i.e.: when $x \vee_S y < 1$, then $\forall a \in [0,1] : a \odot (x \vee_S y) = (a \odot x) \vee_S (a \odot y)$. Without loss of generality, we can assume that $x \leq y$. This implies $x \vee_S y = y < 1$, and $a \odot (x \vee_S y) = a \odot y$. Since $\odot$ is non-decreasing, $(a \odot x) \leq (a \odot y)$, thus $(a \odot x) \vee_S (a \odot y) = a \odot y$, which proves the proposition. □

Among $\vee_S$-type systems, *quasi-Sugeno* systems, for which $\odot = \wedge$, $\wedge$ being a t-norm, play an important role. However, not all t-norms can be used:

Proposition 3. $\mathcal{F} = (\vee_S, \vee_S, \vee_S, \wedge)$*, where* $\wedge$ *is a t-norm, is a t-conorm system for integration if and only if* $\wedge$ *is continuous on* $(0,1]^2$ *and without zero divisors.*

Proof. Recall that an element $a \in (0,1)$ is called a zero divisor of a t-norm $\wedge$ if there exists some $x \in (0,1)$, such that $a \wedge x = 0$. The implication $\Rightarrow$ is trivial. The other implication can be proved using Prop. 2 (with taking in mind that $a \wedge 0 = 0$ for any t-norm $\wedge$). □

For example the Łukasiewicz or the drastic t-norms have zero divisors and hence cannot be used in quasi-Sugeno systems. We can summarize the previous to create the following classification of t-conorm systems for integration:

1. Archimedean systems – can be expressed using Choquet integral, with Choquet integral as a special case.
2. Non-Archimedean systems
   (a) Systems with $\triangle = \underline{\perp}$ – lead to $\vee_S$-type systems, with quasi-Sugeno systems as a special case (and Sugeno integral in particular).
   (b) Systems with $\triangle \neq \underline{\perp}$ – do not express mean value of integrand.

4 Experiments

In this section, we present results of our experiments with quasi-Sugeno t-conorm systems. The first experiment shows the advantage of quasi-Sugeno integral over Sugeno integral, because we can fine-tune the t-norm $\wedge$ for particular data. The second experiment studies performance of quasi-Sugeno integral for "easy" data.

4.1 Experiment 1

In this scenario, we used quasi-Sugeno integral with Aczel-Alsina t-norm, i.e. a t-conorm system $\mathcal{F} = (\vee_S, \vee_S, \vee_S, \wedge_\alpha^{AA})$, and compared its performance for different parameters of the t-norm on two datasets. The Aczel-Alsina t-norm [12] with parameter α is defined as follows:

$$x \wedge_\alpha^{AA} y = \begin{cases} x \wedge_D y & \text{if } \alpha = 0 \\ x \wedge_S y & \text{if } \alpha = \infty \\ \exp(-((-\log x)^\alpha + (-\log y)^\alpha)^{1/\alpha}) & \text{if } \alpha \in (0, \infty) \end{cases} \tag{20}$$

For our experiment, we chose $\alpha = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.2, 1.5, 2, 3$, and $\alpha = \infty$ (which corresponds to the Sugeno integral). Since $\wedge_D$ is not continuous on $(0, 1]^2$, $\alpha = 0$ can not be used. The classifiers $\phi_1, \ldots, \phi_k$ were Bayesian classifiers [13], which used all possible subsets of features (i.e. all possible 1-D Bayesian classifiers, 2-D, 3-D, and so on). The ensemble was aggregated according to the algorithm in Fig. 1.

The results of the testing on two bechmark datasets, Phoneme (from the Elena database, [14]) and Pima (from the UCI repository, [15]) datasets, are shown in Fig. 2 and 3. The figures show average error rates (in %) $\pm$ standard deviations of the aggregated classifiers, measured from 10-fold crossvalidation for different α (solid line). The constant dashed line represents result of the unique, non-combined Bayesian classifier which uses all features.

In the case of the Phoneme dataset, the lowest average error rate was achieved by quasi-Sugeno system with $\wedge_{0.6}^{AA}$ – the improvement over non-combined classifier was about 4% (proved as significant improvement by the two sample t-test against the one-tailed alternative [achieved significance level was 0.00005]). For

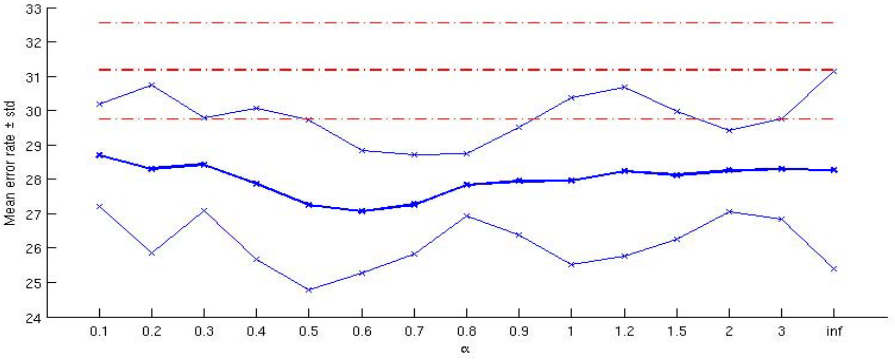


Fig. 2. Results for the Phoneme dataset. Solid line – ensemble aggregated using fuzzy t-conorm integral with $\mathcal{F} = (\vee_S, \vee_S, \vee_S, \wedge_\alpha^{AA})$, dashed line – non-combined classifier.

the Pima dataset, the improvement of about 1%, achieved by $\wedge_{0.7}^{AA}$, did not prove as statistically significant (significance level was 0.12).

What is even more important is that by using t-conorm integral (quasi-Sugeno with Aczel-Alsina t-norm in this case) we can fine-tune the parameters to obtain better results than those of the Sugeno integral. For both datasets, the improvement over Sugeno integral was about 1%, all that achieved without increasing the complexity of the algorithm. Although this improvement did not prove as statistically significant, achieved significance level was 0.15.

4.2 Experiment 2

A common drawback of many methods for combining classifiers is that if the data to classifiy are "easy", then the aggregated classifier often achieves worse results than a simple, non-combined classifier. In this experiment, we will show that for easy data (represented by the 4-dimensional Iris dataset from [15]), we can fine-tune the t-conorm system for integration, so that the performance of the aggregated classifier is only a slightly worse than the performance of non-combined classifier.

We created an ensemble of Bayesian classifiers using all possible subsets of features, resulting in 15 different classifiers. We aggregated the ensemble using quasi-Sugeno t-conorm integral with two different t-norms – Aczel-Alsina (which approaches $\wedge_D$ with $\alpha \to 0$), and Frank t-norm (which approaches $\wedge_L$ with $\alpha \to \infty$), defined as:

$$x \wedge_\alpha^F y = \begin{cases} x \wedge_S y & \text{if } \alpha = 0 \\ x \wedge_P y & \text{if } \alpha = 1 \\ x \wedge_L y & \text{if } \alpha = \infty \\ \log_\alpha \left(1 + \frac{(\alpha^x - 1)(\alpha^y - 1)}{\alpha - 1}\right) & \text{if } \alpha \in (0, \infty), \alpha \neq 1 \end{cases} \tag{21}$$

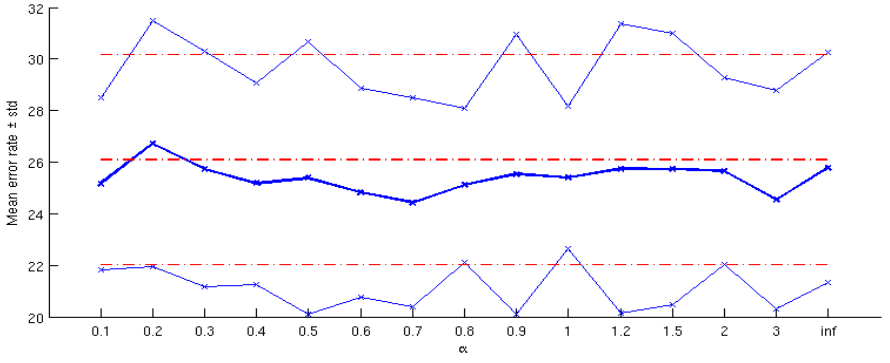


Fig. 3. Results for the Pima dataset. Solid line – ensemble aggregated using fuzzy t-conorm integral with $\mathcal{F} = (\vee_S, \vee_S, \vee_S, \wedge_\alpha^{AA})$, dashed line – non-combined classifier.

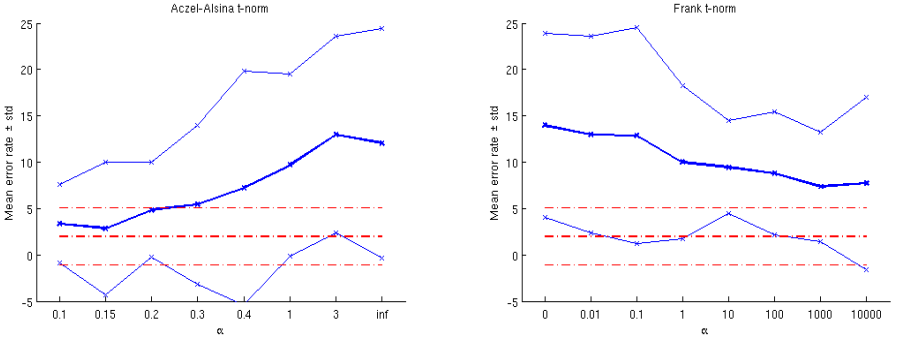


Fig. 4. Results for the Iris dataset. Solid line – ensemble aggregated using fuzzy t-conorm integral with $\mathcal{F} = (\vee_S, \vee_S, \vee_S, \wedge_\alpha^{AA})$ (left) or $\mathcal{F} = (\vee_S, \vee_S, \vee_S, \wedge_\alpha^{F})$ (right), dashed line – non-combined classifier.

In Fig. 4, the results for system with $\wedge_\alpha^{AA}$ for $\alpha = 0.1, 0.15, 0.2, 0.3, 0.4, 1, 3$, and $\alpha = \infty$ (Sugeno integral), and $\wedge_\alpha^{F}$ for $\alpha = 0$ (Sugeno integral), $\alpha = 0.01, 0.1, 1, 10, 100, 1000, 10000$ are shown. We measured average error rates (in %) $\pm$ standard deviations from 10-fold crossvalidation (solid line); the dashed line corresponds to the unique, non-combined Bayesian classifier which uses all features.

The Iris dataset contains only about 150 patterns, so the results of the cross-validation have big variance. We can see that Sugeno integral achieves average error rate about 12-14%. If we use quasi-Sugeno system with Frank t-norm, then as the t-norm approaches $\wedge_L$, the average error rate decreases ($\wedge_L$ cannot be used because it has zero divisors). Even better results were achieved by quasi-Sugeno system with Aczel-Alsina t-norm approaching $\wedge_D$.

5 Summary

In this paper, we described the fuzzy t-conorm integral and its use for combining classifiers. Different classes of t-conorm systems for integration were discussed. We showed that although the framework of fuzzy t-conorm integral is very general, only few t-conorm systems for integration can be used for combining classifiers (although the question which specific t-conorm system to use for a specific application remains unresolved). Still the fuzzy t-conorm integral adds additional degrees of freedom to classifier combining, and so it can provide more succesful way to classifier combining than Sugeno or Choquet fuzzy integral. That was confirmed also by tests on three benchmark datasets.

References

1. Murofushi, T., Sugeno, M.: Fuzzy t-conorm integral with respect to fuzzy measures: Generalization of Sugeno integral and Choquet integral. Fuzzy Sets and Systems 42(1), 57–71 (1991)
2. Melnik, O., Vardi, Y., Zhang, C.H.: Mixed group ranks: Preference and confidence in classifier combination. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(8), 973–981 (2004)
3. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.W.: Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognition 34(2), 299–314 (2001)
4. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
5. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: International Conference on Machine Learning, pp. 148–156 (1996)
6. Bay, S.D.: Nearest neighbor classification from multiple feature subsets. Intelligent Data Analysis 3(3), 191–209 (1999)
7. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. 20(3), 226–239 (1998)
8. Kuncheva, L.I.: Fuzzy versus nonfuzzy in combining classifiers designed by boosting. IEEE Transactions on Fuzzy Systems 11(6), 729–741 (2003)
9. Ahmadzadeh, M.R., Petrou, M.: Use of Dempster-Shafer theory to combine classifiers which use different class boundaries. Pattern Anal. Appl. 6(1), 41–46 (2003)
10. Grabisch, M., Nguyen, H.T.: Fundamentals of Uncertainty Calculi with Applications to Fuzzy Inference. Kluwer Academic Publishers, Norwell (1994)
11. Chiang, J.H.: Aggregating membership values by a Choquet-fuzzy-integral based operator. Fuzzy Sets Syst. 114(3), 367–375 (2000)
12. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Kluwer Academic Publishers, Dordrecht (2000)
13. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, Chichester (2000)
14. Elena database: `http://www.dice.ucl.ac.be/mlg/?page=Elena`
15. Newman, D.J., Hettich, S., Merz, C.B.: UCI repository of machine learning databases (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`

Integrated Query Answering with Weighted Fuzzy Rules

Alexandros Chortaras*, Giorgos Stamou, and Andreas Stafylopatis

School of Electrical and Computer Engineering
National Technical University of Athens
Zografou 157 80, Athens, Greece
{achort, gstam}@softlab.ntua.gr, andreas@cs.ntua.gr

Abstract. Weighted fuzzy logic programs increase the expressivity of fuzzy logic programs by allowing the association of a significance weight with each atom in the body of a fuzzy rule. In this paper, we propose a prototype system for the practical integration of weighted fuzzy logic programs with relational database systems in order to provide efficient query answering services. In the system, a dynamic weighted fuzzy logic program is a set of rules together with a set of database queries, fuzzification transformations and fact derivation rules, which allow the provided set of rules to be augmented with a set of fuzzy facts retrieved from the underlying databases. The weights of the rules may be estimated by a neural network-based machine learning process using some specially designated for this purpose training database data.

1 Introduction

The handling of uncertainty is an important requirement for logic programming because in some applications it is difficult to determine the precise truth value of facts. Using fuzzy logic, logic programming can be provided with the ability to reason with uncertain knowledge (e.g. [14], [9], [6]). *Weighted fuzzy logic programs* [4], [2] are a further extension of fuzzy logic programming frameworks which, by introducing significance weights, allow each atom in a rule body to have a different importance. For example, the rule $1.0 : Happy(x) \leftarrow \widetilde{\wedge}((0.3; Rich(x)), (0.8; Healthy(x)))$ may model the fact that health is more important than wealth for the happiness of a person. The semantics of weighted fuzzy logic programs have been studied within the model-theoretic and fixpoint paradigms. Query answering is more elaborate than in classical logic because all the alternative ways of inferencing a fact must be considered [9]. Based on resolution and tabling methods [12], [5], a sound and complete top-down query answering procedure for weighted fuzzy logic programs is outlined in [3].

* A. Chortaras is on a scholarship from the Onassis Public Benefit Foundation. This work is partially funded by the X-Media and ASSIST projects, sponsored by the European Comission under contract no. IST-FP6-26978 and IST-2002-027510 respectively.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 767–778, 2007.

Despite the increased expressivity of weighted fuzzy logic programs, their usefulness can be exploited in practice only if they can be integrated with stores of already existing knowledge so as to provide an additional inference service layer upon them. In classical logic, database systems equipped with a rule language and inference engine are known as *deductive databases* and *datalog* has been used as the rule language, whose semantic relation with relational algebra is a well-studied problem [13]. Because deductive databases are a mature field, even in recent ontology-based systems, it is desirable to reduce an ontology-based knowledge base to a (disjunctive) datalog program (e.g. [11], [7]), so that query answering over the ontology can be performed using well-established deductive database techniques.

Another important issue for the use of weighted fuzzy logic programs in such a setting is the availability of techniques for computing the weights of the rules so as to obtain programs that effectively describe the underlying knowledge.

Based on the two above-outlined directions, we propose a practical framework for the integration of weighted fuzzy logic programs with relational database systems, so that the databases may be used as the source of ground facts, on the one hand for learning the weights and on the other for performing the knowledge inferencing procedure modeled by the rules of a weighted fuzzy logic program. The aim is to achieve this integration in a way as transparent as possible for the user and without resorting to any intervention in the databases. The final goal is to provide an effective query answering service over the integrated adaptable weighted fuzzy logic program and database systems.

The rest of the paper is as follows: section 2 provides an overview of weighted fuzzy logic programs, section 3 presents the way the database integration is achieved, section 4 the overall system architecture and section 5 discusses the rule adaptation process. Section 6 outlines an example use case and section 7 concludes the paper.

2 Weighted Fuzzy Logic Programs

2.1 Syntax and Semantics

A *fuzzy atom* is the formula $p(u_1, \ldots, u_n)$, where p is a fuzzy predicate and each u_i a variable or a constant. The *truth* and *falsehood* fuzzy atoms t and f are special atoms that represent absolute truth and falsehood respectively. A fuzzy atom A is *subsumed* by the fuzzy atom B ($A \sqsubseteq B$) if there exists a substitution of variables θ such that $A = B\theta$.

A *weighted fuzzy logic program* is a finite set of rules of the form

$$w : B \leftarrow \widetilde{\wedge}((w_1; A_1), \ldots, (w_n; A_n))$$

where B is a fuzzy atom excluding t and f, such that each variable that appears in B appears also in at least one of the atoms A_i, i.e. the rule is *safe*. $w \in [0, 1]$ is the *strength* of the rule and each $w_i \in [0, 1]$ models the *significance* of A_i.

A fuzzy atom or rule that does not contain variables is *ground* and a rule whose body includes only the truth and falsehood fuzzy atoms is a *fuzzy fact*.

For convenience, for a rule R we write also $s(R)$ instead of w, $c(R)$ instead of B, $\boldsymbol{w}(R)$ for the vector $(w_1, \ldots, w_n)$ and $\boldsymbol{a}(R)$ for the vector $(A_1, \ldots, A_n)$. The body of R is a *weighted fuzzy conjunction*, i.e. a conjunction in which the higher the weight associated with an element is, the more that element affects the value of the conjunction. We evaluate such a conjunction using a *weighted conjunction operator* $\widetilde{\wedge}_{[\cdot]}$, whose properties are presented in [4].

Let $\mathcal{P}$ be a weighted fuzzy logic program, $U_{\mathcal{P}}$ its *Herbrand universe* (the set of constants of $\mathcal{P}$) and $B_{\mathcal{P}}$ its *Herbrand base* (the set of ground atoms constructed from the predicates of $\mathcal{P}$ and constants in $U_{\mathcal{P}}$). The *inference base* $R_{\underline{B}}$ of a ground fuzzy atom $\underline{B}$ is the set of all the ground rule instances of $\mathcal{P}$ that consist of atoms from $B_{\mathcal{P}}$ and have $\underline{B}$ as their head. A *fuzzy interpretation* of $\mathcal{P}$ is then a mapping that maps each atom in $B_{\mathcal{P}}$ to a truth value. The truth fuzzy atom is mapped to 1, and the falsehood fuzzy atom to 0. The interpretation mapping all atoms (except t) to 0 is denoted by $I_{\perp}$.

We denote a ground rule by $\underline{R}$, a ground fuzzy atom by $\underline{A}$, and by $\underline{A}^I$ its truth value under I. If $\underline{R}$ is the rule $w : \underline{B} \leftarrow \widetilde{\wedge}((w_1; \underline{A}_1), \ldots, (w_n; \underline{A}_n))$ and I an interpretation, we write also $\boldsymbol{a}^I(\underline{R})$ for the vector $(\underline{A}_1^I, \ldots, \underline{A}_n^I)$ and $c^I(\underline{R})$ instead of $\underline{B}^I$.

Given a weighted conjunction operator $\widetilde{\wedge}_{[\cdot]}$, a t-norm T and an s-norm S (for details see e.g. [8]), the interpretation I is a *model* of the weighted fuzzy logic program $\mathcal{P}$ under $(\widetilde{\wedge}_{[\cdot]}, T, S)$, if for all fuzzy ground atoms $\underline{B} \in B_{\mathcal{P}}$ we have (cf. [9], [6])

$$\underline{B}^{FT_{\mathcal{P}}(I)} \doteq S\left(\left\{T(\widetilde{\wedge}_{[\boldsymbol{w}(R)]}(\boldsymbol{a}^I(\underline{R})), s(R))\right\}_{\underline{R} \in R_{\underline{B}}}\right) \leq \underline{B}^I$$

The operator $FT_{\mathcal{P}}$ defined in the above is the *fuzzy immediate consequence operator*, which is continuous if $\widetilde{\wedge}_{[\cdot]}$, T, S are continuous. According to fixpoint theory $FT_{\mathcal{P}}$ has a least fixpoint, which, if $FT_{\mathcal{P}}$ is continuous, can be determined by a process of ω iterative applications of $FT_{\mathcal{P}}$ starting from $I_{\perp}$. It follows that $\mathcal{P}$ has a unique *minimal model* $FM_{\mathcal{P}}$, defined as the *intended meaning* of $\mathcal{P}$ under $(\widetilde{\wedge}_{[\cdot]}, T, S)$ and if $FT_{\mathcal{P}}$ is continuous we have that $FM_{\mathcal{P}} = {FT_{\mathcal{P}}}^{\uparrow\omega}$.

In several cases (e.g. if $\mathcal{P}$ is has no recursive rules), ${FT_{\mathcal{P}}}^{\uparrow\omega}$ may be reached in a finite number of iterations. In general it may be impossible to reach ${FT_{\mathcal{P}}}^{\uparrow\omega}$ in less than ω steps, in which case the semantics of $\mathcal{P}$ are non-computable. To overcome this difficulty we rely on a *fixpoint approximation* which is always computable. To quantify the approximation we need an interpretation *distance metric*. Given such a metric ρ, it can be shown (cf. [14]) that if $FT_{\mathcal{P}}$ is continuous, either there exists a k such that $FM_{\mathcal{P}} = {FT_{\mathcal{P}}}^{\uparrow k}$, or for any $\epsilon > 0$ there exists a k such that $\rho(FM_{\mathcal{P}}, {FT_{\mathcal{P}}}^{\uparrow n}) < \epsilon$ for $n > k$. Hence, an approximation of ${FT_{\mathcal{P}}}^{\uparrow\omega}$ at any level of accuracy is computable in $k < \omega$ steps. The relation between ϵ and k is in general not known; for this reason, a practical criterion is to stop the iteration whenever for some n and sufficiently low $\epsilon' > 0$ we get $\rho({FT_{\mathcal{P}}}^{\uparrow n+1}, {FT_{\mathcal{P}}}^{\uparrow n}) < \epsilon'$.

2.2 Query Answering

Query answering is performed by a computational procedure which, given a weighted fuzzy logic program $\mathcal{P}$ and a fuzzy atom A, computes the answer to

the query $?A$ by finding all the ground fuzzy atoms that are subsumed by A and have a non-zero truth value in $FM_{\mathcal{P}}$. Given the preceding discussion we may be able to answer a query only approximately: if $\underline{A} \sqsubseteq A$, then $(\underline{A}, v)$ is an ϵ-*correct* answer to $?A$ for some $\epsilon > 0$ if $\underline{A}^{FM_{\mathcal{P}}} < v + \epsilon$ and a *correct* answer if $\underline{A}^{FM_{\mathcal{P}}} = v$.

In order to perform query answering with weighted fuzzy logic programs we construct a *resolution graph* which models in a directed graph the dependencies between the query and the program rules. The graph has a *skeleton*, i.e. a tree that encodes the non-recursive rule dependencies. The tree is augmented into a cyclic graph if the program contains recursive rules. Using the resolution graph structure, [3] outlines a sound and complete query-answering algorithm for weighted fuzzy logic programs. The algorithm terminates always by computing an ϵ-correct answer if a correct answer is not computable in finite time and holds the intermediate results, that allow it to avoid entering into infinite chains of computations, within the resolution graph.

The resolution graph G is constructed by creating first the root of its skeleton, which coincides with A, and then by adding step by step new edges and nodes by computing the inference base of each intermediate goal. The nodes are either atom or rule nodes. An atom node is labeled by a fuzzy atom and a rule node by a rule of the $\mathcal{P}$ possibly with some of its variables bound to constants. Each atom node s has as child one rule node for each element in the inference base of its label, and a rule node has one atom node child for each atom in its body. If an atom node s is added in the graph whose label is subsumed by an already existing atom node s' in G, s is linked by an edge with s' and no rule nodes are added as its children. Each atom node with label the fuzzy atom A maintains also a set of substitution-value pairs (θ, v) which store the truth value v (if $v > 0$) that has been computed for the ground fuzzy atom $A\theta$.

Given an initial interpretation, G is evaluated bottom-up, starting from the leaves of its skeleton and proceeding towards the root. During this process, the substitution-value pairs for each node are computed by considering the already completed computations of the children. If G is acyclic this process terminates once the skeleton root is reached. Otherwise, an iterative evaluation of G may be necessary in order to reach, or approximate the query answer. It can be proved that the process is sound and complete so that for any substitution θ such that $\underline{A} = A\theta$ there is a resolution graph G and an iterative valuation of it starting from $I_{\perp}$ such that $(\underline{A}, v)$ is either a correct or an ϵ-correct answer to $?A$ for any $\epsilon > 0$, where v is the value held in the root of G for substitution θ.

3 Databases and Fact Derivation

Following the classical logic programming paradigm, a weighted fuzzy logic program $\mathcal{P}$ does no distinction between facts and rules, since the first are only a special case of the second. In a database-oriented approach however it is useful to be able to distinguish between facts and rules, since the same rules may be used in combination with different data. Moreover, the data may change during the lifetime of the databases without that the rules that describe them change.

It is then desirable that the inference is always performed based on the actual data present in the databases at the time the query is posed.

This requirement precludes the use of architectures in which *exhaustive forward inference* is performed at the time the data is loaded from the databases (e.g. [1] in the context of RDF querying for the Semantic Web). These systems typically create a knowledge repository which stores both the explicit in the databases knowledge and the implicit knowledge inferred from the provided rule set. Such a system can respond very quickly to a query since it has precomputed all the answers, but if the underlying databases are modified, the knowledge repository needs to be updated or re-computed, which is in general a high cost task.

As a result, our attempt here is to disentangle the rules from the facts of a weighted fuzzy logic program, isolate the set of rules, and be able to define and dynamically attach to it sets of facts obtained from some underlying databases. Recall that a fuzzy fact is a rule of the form $w : \underline{A} \leftarrow \widetilde{\wedge}((w_1; u), \ldots, (w_n; u))$ where u is either t or f. However, a weight $w' = f(w, w_1, \ldots, w_n)$ for some function f may be computed, for which the above is equivalent to $w' : \underline{A} \leftarrow (1; t)$, the preferred way to express a fact.

Given a database, our first aim is therefore to define a process that generates from it a set of facts, semantically equivalent to fuzzy facts of the form $v : \underline{A} \leftarrow (1; t)$. This process involves two main steps: a) the definition and execution of a database query Q, and b) the processing of the result of Q, the construction out of it of one or more fuzzy ground atoms $\underline{A}$ and the computation for each one of them of a truth value v in the interval $[0, 1]$. The second step in most cases will involve a fuzzification of some crisp data obtained from the database.

The exact queries that should be executed on the databases and the subsequent transformations that the query results should undergo in order to produce the desirable set of fuzzy facts is a procedure that must be manually crafted. For this reason, in order to perform the fact derivation, the system we propose relies on the first place on the existence of a set of database queries and fuzzifiers. Making use of the definitions included in these sets, fact derivation rule sets for the actual generation of the facts may then be defined. We describe now the proposed language in which these sets can be defined.

Database Queries: Since we base our system on the relational database model, the set of queries must be a set of queries expressed in the appropriate SQL dialect supported by the underlying databases. Each query is defined as

$$Query[Database] = SQLExpression$$

This statement defines the query $SQLExpression$, which should be executed on the database named $Database$. The query is assigned the name $Query$. The result of the execution of $SQLExpression$ can be regarded as a table whose columns are defined in the $SELECT$ part of $SQLExpression$. The i-the column of the resulting table can be referenced by $<Query : i>$, or $<Query : Column>$ if $Column$ is the name of the i-th column.

Fuzzifiers: A fuzzifier is a function that can be used for the fuzzification of any crisp data obtained from the database. A fuzzifier is defined as a branched

function

$$Name(Var) = \{\ Condition_1 : Expression_1 \mid \ldots \mid Condition_n : Expression_n\}$$

where $Name$ is the name of the fuzzifier, Var a variable name, and $Condition_i$ a condition for Var that if satisfied, the computed value for Var is the result of the evaluation of the mathematical expression $Expression_i$. The expressivity allowed in $Expression_i$ depends on the implementation; in all cases it should involve only the variable Var.

Fuzzy Fact Derivation Rules: Given a set of database queries $\mathcal{Q}$ and a set of fuzzifiers $\mathcal{F}$, we define a set of database-dependent facts using expressions of the form

$$Predicate(List) = Fuzzifier(Column)$$

where $Predicate$ is the name of the fuzzy predicate for the newly constructed facts, $List$ is a comma separated list of constants or $Column$s, and $Fuzzifier$ is the name of a fuzzifier in $\mathcal{F}$. $Column$ is a column of the result of a database query in $\mathcal{Q}$. As we have already seen, if $Query$ is a database query, $<Query : i>$ refers to the i-th column of the results table. In each fact definition, the $Column$ argument of $Fuzzifier$ and all the $Column$s in $List$ must refer to the same query. Such a statement defines a set of facts: once the $Query$ included in $List$ and/or $Column$ is executed against the database, for each row of the resulting table, a new fuzzy fact will be constructed, such that the arguments of $Predicate$ and $Fuzzyfier$ are replaced by the respective column entries. More formally, for the statement

$$Predicate(Ex_1, \ldots, Ex_n) = Fuzzifier(<Query : j>)$$

where Ex_i is either $< Query : p_i >$ or a constant c_i, for each row r in the result table of $Query$, the fuzzy fact

$$val_j : Predicate(arg_1, \ldots, arg_n) \leftarrow (1; t)$$

is constructed, where $val_j = Fuzzifier(<Query : j>_r)$, $arg_i = <Query : p_i>_r$ or $arg_i = c_i$, and $<Query : j>_r$ is the value in the j-th column of row r. In practice, a new fact needs to be constructed only if $val_j > 0$. In the case that no fuzzifier is needed, the simpler expression $Predicate(List) = Value$ will simply assign the numerical value $Value$ as strength to all the constructed fuzzy facts.

Weighted Fuzzy Rules: In order to complete the presentation of the language used in the system, we present also the syntax for writing weighted fuzzy rules. The statement

$$s : Head(List) :- w_1 : Body_1(List_1)\ \&\ \ldots\ \&\ w_n : Body_n(List_n)$$

defines the rule $s : Head(List) \leftarrow \widetilde{\wedge}((w_1; Body_1(List_1)), \ldots, (w_n; Body_n(List_n)))$, where $List$ and $List_i$ are comma separated lists of constants and variables such that the rule is safe. Variable is considered to be any element that starts with an underscore. The strength of the rule, or any of the weights in the body can be omitted if they are equal to 1.

4 System Architecture

From the above discussion, it follows that a weighted fuzzy logic program $\mathcal{P}$ (which for convenience we will call also a *set of weighted fuzzy rules* in the sequel) augmented with a set of database queries $\mathcal{Q}$, a set of fuzzifiers $\mathcal{F}$ and of fact derivations rules $\mathcal{E}$ is equivalent, for a given instance of the databases involved in the elements of $\mathcal{Q}$, to a dynamic weighted fuzzy logic program $\mathcal{P}^*$ which consists of the rules in $\mathcal{P}$ and the facts derived from $\mathcal{E}$ as described before. Hence, $\mathcal{P}^*$ may be defined as the tuple $(\mathcal{P}, \mathcal{Q}, \mathcal{F}, \mathcal{E})$. For practical reasons, and because the same sets of queries, fuzzifiers may be used in several fact derivation rule sets, and in its turn a set of fuzzy rules may be used in combination with several fact derivation rule sets, it is useful to keep the several sets distinct, in the form of libraries, and combine them each time as desired into new, custom, *dynamic* programs. This is the approach followed by our system, which offers a query-answering service on weighted fuzzy logic programs built upon a client-server architecture.

The server is responsible for keeping and informing the client about the available sets $\mathcal{Q}$, $\mathcal{F}$, $\hat{\mathcal{E}}$ and $\hat{\mathcal{P}}$, where $\mathcal{Q}$, $\mathcal{F}$ are as before, and $\hat{\mathcal{E}}$ is the set of fuzzy fact derivation rule sets and $\hat{\mathcal{P}}$ a set of weighted fuzzy rule sets. A user may then ask a query on a dynamic weighted fuzzy logic program by sending to the server a message of the form

$$QUERY\ (WC, T, S, epsilon)\ RuleSet\ FactSet1, ..., FactSetN\ Query$$

The tuple $(WC, T, S, epsilon)$ contains the parameter values that will be used for the computation of the answer: the first three correspond to the triple of operators $(\widetilde{\wedge}_{[\cdot]}, T, S)$ under which the semantics will be computed and *epsilon* defines the level of desired accuracy in the case of a recursive program with non-computable semantics. The available values for the operators WC, T, S may be obtained from the user by asking the server. $RuleSet$ and $FactSet1, ..., FactSetN$ define the actual dynamic program on which the query $Query$ is posed. $RuleSet$ must belong to $\hat{\mathcal{P}}$ and $FactSet1, ..., FactSetN$ to $\hat{\mathcal{E}}$ and determine the fuzzy facts with which $RuleSet$ will be extended. $Query$ is an expression of the form $Predicate(List)$, where $List$ is a comma separated list of variables (anything that starts with an underscore) or constants. As a response to the query, once the computation has been completed, the client will receive a set of $(Predicate(List'), v)$ statements, which includes all the (ϵ-)correct answers to the query for which $v > 0$ and $Predicate(List')$ is a ground fuzzy atom subsumed by $Predicate(List)$.

The server is made up from four main modules: The data library, the database translation, the reasoning and the rule learning module. The rule learning will be the subject of the next section. The data library module is the one that holds the sets $\mathcal{Q}$, $\mathcal{F}$, $\hat{\mathcal{E}}$ and $\hat{\mathcal{P}}$. The translation module is responsible for communicating with the underlying databases and deriving the necessary facts defined in $\hat{\mathcal{E}}$. Finally, the reasoning module is the module that for each user query dynamically constructs and evaluates the corresponding resolution graph so as to correctly answer the query.

As we have mentioned before, a pre-requisite for the system is that it should answer each query by performing inference using the actual data present in the database at the time of the query. This implies that each answering of a query involves a number of databases accesses. A central performance issue for the proposed system is therefore to determine the minimal number of database accesses and amount of loaded data that is necessary to correctly answer a given query. The naïve solution for the server is to get the $FactSet1, ..., FactSetN$ fact derivation rule sets, construct indiscriminately, by accessing the databases, all the fuzzy facts defined in them, augment $RuleSet$ with the new facts and then execute the resolution algorithm on the augmented program, which is now self-dependent and does not involve any database accesses. This is obviously not an optimal approach because only a small part of the constructed facts may in fact be needed to answer the query. A more efficient strategy for the server would be to start the construction of the resolution graph for $Query$ and whenever an atom node is added in it with a predicate in its label that is defined in the fact derivation rule sets, dynamically access the database in order to retrieve and construct only the relevant facts.

In most cases this will be an attainable strategy. There is however a theoretical complication, which a very simple example will illustrate.

Consider the set $\mathcal{P}$ of weighted fuzzy rules that consists of the two rules

$$1 : p(x) \leftarrow \widetilde{\wedge}((0.8; r(x, y)), (0.2; s(y, z))) \text{ and } 1 : q(x) \leftarrow (1; u(x))$$

and that the fact derivation rule set $\mathcal{E}$ contains statements for the construction of facts for the predicates r, s and u. Assume that for the evaluation of the semantics a triple of operators is used $(\widetilde{\wedge}_{[\cdot]}, T, S)$ such that S is the Łukasiewicz s-norm $S_{luk}(a_1, a_2) = \min\{1, a_1 + a_2\}$ and suppose also that the facts that can be derived from the database are only

$$0.5 : r(a, b) \leftarrow (1; t) \text{ and } 1 : u(c) \leftarrow (1; t)$$

It follows that the Herbrand universe of the resulting dynamic program (i.e. of the program $\mathcal{P}$ augmented with the facts constructed using the rules in $\mathcal{E}$) is the set $\{a, b, c\}$ and as a result, according to the definition of the minimal model of a weighted fuzzy logic program, the correct answer to the query $?p(a)$ is $(p(a), v)$ where v is the Łukasiewicz s-norm of the values of the three weighted conjunctions $\widetilde{\wedge}((0.8; r(a, b)), (0.2; s(b, a)))$, $\widetilde{\wedge}((0.8; r(a, b)), (0.2; s(b, b)))$ and $\widetilde{\wedge}((0.8; r(a, b)), (0.2; s(b, c)))$. However, the resolution graph built for $?p(a)$ will consist of a root node for $p(a)$ and at the next level of atom nodes it will have one atom node for $r(a, y)$ and one for $s(y, z)$. Since the definitions for the predicates r and s are in $\mathcal{E}$ the database access and fact derivation mechanism will be invoked and as expected will generate only a fact for $r(a, b)$. The existence of the constant c is not known to the graph, hence the query will be answered incorrectly by computing the s-norm of the values of the weighted conjunctions $\widetilde{\wedge}((0.8; r(a, b)), (0.2; s(b, a)))$ and $\widetilde{\wedge}((0.8; r(a, b)), (0.2; s(b, b)))$ only.

The problem is clearly that in general in order to compute the answer to the query, the entire Herbrand universe must be known, which in our case translates into constructing using the fact derivation rule sets all the possible facts

regardless of whether they 'seem' to be related with the query or not. Of course this is an artificial example, which is unlikely to be encountered in real applications with well-designed databases and rule sets. However it demonstrates that it is essential for the system to be able to determine (by a static analysis of the program) whether it needs first to construct the entire set of facts in order to answer a query or it can construct them only dynamically during the query-answering process. It is easy to see that if every variable that appears in a rule body appears in at least one atom with the highest weight in the rule or in the rule head, the above-described problem cannot occur and facts can always be retrieved dynamically as needed from the database.

5 Rule Learning

In the preceding discussion, we have assumed that the rule sets in $\hat{\mathcal{P}}$ are all known and well-defined. In general, it is expected that these rule sets will be defined by domain expects. However, it is quite unrealistic to expect that the great number of parameters involved in the rules can be precisely and correctly determined using the a priori domain knowledge of the experts. A rough estimate for the values of the weights may in general be available, which will need to be adapted using some training data, so that the logical theory represented by a weighted fuzzy logic program describes well the actual data.

Machine learning techniques are useful in this respect and a neural network based technique for the adaptation of the weights of weighted fuzzy logic programs is introduced in [2]. The adaptation is based on the minimization of the square error between the truth value contained in a training dataset for a certain fuzzy atom, and the truth value computed for that atom by the rules of a weighted fuzzy logic program. The minimization is performed by considering independently each predicate p that appears in a weighted fuzzy logic program. In particular, if B_p is the subset of the Herbrand base of a weighted fuzzy logic program $\mathcal{P}$ that contains the ground atoms of predicate p, the optimal weights of the rules that have in their heads an atom of predicate p are the ones that minimize the quantity

$$\sum_{\underline{B}\in B_p}\left(S\left(\left\{T(\widetilde{\wedge}_{[\boldsymbol{w}(R)]}(\boldsymbol{a}^I(\underline{R})),s(R))\right\}_{\underline{R}\in R_{\underline{B}}}\right)-\underline{B}^I\right)^2 \tag{1}$$

with respect to the weights $\boldsymbol{w}(R)$ and $s(R)$. In order to use this expression however, we need to determine the interpretation I, which serves as the source of the training data. This is straightforward using the technique described in section 3. Since I may be seen as a program $\mathcal{F}_I$ containing only facts, in particular a fact $v:\underline{B}\leftarrow(1;t)$ if I maps $\underline{B}$ to a $v>0$, we can define I as a database derivable set of facts, using a specially designated for this purpose database. Since the minimal model of $\mathcal{F}_I$ is I, having determined the program $\mathcal{F}_I$ we have determined also the training interpretation I and so we can use the above expression.

The computation of the weights of each rule in a rule set of $\hat{\mathcal{P}}$ using the above optimization process, is performed by the learning module of the server, which

uses the functionality of the database translation module in order to construct the training dataset described by $\mathcal{F}_I$. The learning is performed for each rule set that is added to the server, using a designated set of database data in order to derive the training facts following the rules in $\mathcal{F}_I$. We have to note that in order to perform the training, $\mathcal{F}_I$ must be a complete set of facts in the sense that it must contain a fact derivation rule for each fuzzy atom that appears in the body or the head of each rule we want to learn, so that (1) is well-defined. This is a property that the designated for the training process dataset must have and not the databases on which the inference will subsequently be performed, since the role of the learned rules is precisely to apply the learned logical relations on actual databases in order to draw new inferences.

6 Example

In order to illustrate the use of the system we outline a use case based on data provided by the ASSIST project, which deals with the provision to medical researchers of an environment to study cervical cancer based on patient records.

The available database has the form of a table with clinical data for several patients. Among others, for each patient several personal data (e.g. age, ethnic origin, number of children), health related information (e.g. years of smoking), results of clinical tests (e.g, cytology, colposcopy, histology) and related genetical information are kept.

Some examples of database queries, fuzzifiers, fact derivation rules as well as of weighted fuzzy rules for this database are presented in figures 1-4. Based on them, an example query is shown in figure 5. Note that the provided statements are only fragmentary and their aim to illustrate the use and expressive capabilities of the system.

```
Q_PatientData[ASSIST_DB] =
    SELECT Vid, Age, Cytology, Colposcopy, Histology FROM Patients;
Q_EthnicOriginCaucasian[ASSIST_DB] =
    SELECT Vid FROM Patients WHERE ETHNIC_ORIGINE='C';
Q_EthnicOriginAsian[ASSIST_DB] =
    SELECT Vid FROM Patients WHERE ETHNIC_ORIGINE='A';
```

Fig. 1. Some database queries

```
f_AgeYoung(x) = { x<25 : 1 |  25<=x && x<=50 : -1/25*(x-25)+1 |  x>50 : 0 }
f_AgeOld(x) = { x<50 : 0 |  50<=x && x<=75 : 1/25*(x-50) |  x>75 : 1 }
f_ThreeValueFuzzifier(x) = { x == 0 : 0 |  x == 1 : 0.33 |  x == 2 : 0.67 |
                             x == 3 : 1 }
```

Fig. 2. Some fuzzifiers

```
Young(<Q_PatientData:Vid>) = f_AgeYoung(<Q_PatientData:Age>)
Old(<Q_PatientData:Vid>) = f_AgeOld(<Q_PatientData:Age>)
Caucasian(<Q_EthnicOriginCaucasian:Vid>) = 1
HistologyResult(<Q_PatientData:Vid>) =
    f_ThreeValueFuzzifier(<Q_PatientData:Histology>)
ColposcopyResult(<Q_PatientData:Vid>) =
    f_ThreeValueFuzzifier(<Q_PatientData:Colposcopy>)
```

Fig. 3. Some fuzzy fact derivation rules from ASSIST_Facts

```
%Highly abnormal histology and pap test results are indicative of invasive carcinoma
1.0 : InvasiveCarcinoma(_x) :- HistologyResult(_x)
1.0 : InvasiveCarcinoma(_x) :- CytologyResult(_x)
%Highly abnormal colposcopy results are a suspicion of invasion
0.6 : InvasiveCarcinoma(_x) :- ColposcopyResult(_x)
%Mildly abnormal pap test results are thought to be worrisome only if HPV is detected
Alarm(_x) :- 0.7: CytologyResult(_x) & HPVInfection(_x)
```

Fig. 4. Some weighted fuzzy rules from ASSIST_Program

```
QUERY (standard, minimum, maximum, 1e-10)
            ASSIST_Program ASSIST_Facts Alarm(_x)
```

Fig. 5. A query message

7 Conclusions

We have introduced a prototype system for the provision of inference and query answering services based on the integration of weighted fuzzy logic programs with relational database systems. The weighted fuzzy logic program models the knowledge inference process and the databases act as the source of ground knowledge for the inferencing and for the learning of the rule weights. A complete environment for defining the fact derivation process has been presented and to facilitate its deployment the system has been built upon the client-server architecture.

Regarding future work, the system needs still to be tested with large scale real data in order to assess its performance. At the implementation level, the optimization of the resolution and database access algorithms need a detailed study since it is clear that the practical usefulness of the system depends on the efficiency of the implementation. Some optimization issues have already been raised in the context of this paper.

At the semantic level, the main limitation of the system is that it is restricted to rules without negation. We are currently working on the development of a resolution process for programs that allow the use of negation as failure. This extension will greatly enhance the expressivity of weighed fuzzy logic programs and increase their potential to be used with real applications. The handling of

missing information in the underlying databases is also a crucial issue. Currently the system relies on the Closed World Assumption, which may be inappropriate in several cases. The Kripke-Kleene or the Any-Word Assumption semantics [10] are alternatives to be considered.

References

1. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A generic architecture for storing and querying RDF. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 54–68. Springer, Heidelberg (2002)
2. Chortaras, A., Stamou, G., Stafylopatis, A.: Adaptation of weighted fuzzy programs. In: Kollias, S., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4132, pp. 45–54. Springer, Heidelberg (2006)
3. Chortaras, A., Stamou, G., Stafylopatis, A.: Top-down computation of the semantics of weighted fuzzy logic programs. In: The 1st Int. Conf. on Web Reasoning and Rule Systems (to appear)
4. Chortaras, A., Stamou, G., Stafylopatis, A., Kollias, S.: A connectionist model for weighted fuzzy programs. In: IJCNN'06: Int. Joint Conf. on Neural Networks, pp. 5362–5369 (2006)
5. Damásio, C.V., Medina, J., Ojeda Aciego, M.: A tabulation proof procedure for first-order residuated logic programs: Soundness, completeness and optimizations. In: FUZZ'06: Int. Conf. on Fuzzy Systems, pp. 9576–9583 (2006)
6. Damásio, C.V., Pereira, L.M.: Sorted monotonic logic programs and their embeddings. In: IPMU'04: Information Processing and Management of Uncertainty, pp. 807–814 (2004)
7. Hustadt, U., Motik, B., Sattler, U.: Reducing SHIQ$^-$ description logic to disjunctive datalog programs. In: KR2004: 9th Int. Conf. on Knowledge Representation and Reasoning, pp. 152–162 (2004)
8. Klir, G., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice-Hall, Upper Saddle River (1995)
9. Lakshmanan, L.V.S., Shiri, N.: A parametric approach to deductive databases with uncertainty. IEEE Transactions on Knowledge and Data Engineering 13(4), 554–570 (2001)
10. Loyer, Y., Straccia, U.: Any-world assumptions in logic programming. Theoretical Computer Science 342(2-3), 351–381 (2005)
11. Motik, B., Sattler, U., Studer, R.: Query answering for OWL-DL with rules. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 549–563. Springer, Heidelberg (2004)
12. Tamaki, H., Sato, T.: OLD resolution with tabulation. In: Shapiro, E. (ed.) Third International Conference on Logic Programming. LNCS, vol. 225, pp. 84–98. Springer, Heidelberg (1986)
13. Ullman, J.D.: Principles of Database and Knowledge-base Systems. Computer Science Press (1995)
14. Vojtáš, P.: Fuzzy logic programming. Fuzzy Sets and Systems 124, 361–370 (2001)

On Decision Support Under Risk by the WOWA Optimization*

Włodzimierz Ogryczak and Tomasz Śliwiński

Warsaw University of Technology, Institute of Control & Computation Engineering,
00-665 Warsaw, Poland
{wogrycza,tsliwins}@ia.pw.edu.pl

Abstract. The problem of averaging outcomes under several scenarios to form overall objective functions is of considerable importance in decision support under uncertainty. The fuzzy operator defined as the so-called Weighted OWA (WOWA) aggregation offers a well-suited approach to this problem. The WOWA aggregation, similar to the classical ordered weighted averaging (OWA), uses the preferential weights assigned to the ordered values (i.e. to the worst value, the second worst and so on) rather than to the specific criteria. This allows one to model various preferences with respect to the risk. Simultaneously, importance weighting of scenarios can be introduced. In this paper we analyze solution procedures for optimization problems with the WOWA objective function. A linear programming formulation is introduced for optimization of the WOWA objective with monotonic preferential weights. Its computational efficiency is analyzed.

1 Introduction

Consider a decision problem under uncertainty where the decision is based on the maximization of a scalar (real valued) outcome. The final outcome is uncertain and only its realizations under various scenarios are known. Exactly, for each scenario S_i $(i = 1, \ldots, m)$ the corresponding outcome realization is given as a function of the decision variables $y_i = f_i(\mathbf{x})$. We are interested in larger outcomes under each scenario. Hence, the decision under uncertainty can be considered a multiple criteria optimization problem:

$$\max \{ \ (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})) \ : \ \mathbf{x} \in \mathcal{F} \ \} \tag{1}$$

where $\mathbf{x}$ denotes a vector of decision variables to be selected within the feasible set $\mathcal{F} \subset R^q$, of constraints under consideration and $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ is a vector function that maps the feasible set $\mathcal{F}$ into the criterion space R^m.

* The research was supported by the Ministry of Science and Information Society Technologies under grant 3T11C 005 27 "Models and Algorithms for Efficient and Fair Resource Allocation in Complex Systems."

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 779–790, 2007.

From the perspective of decisions under uncertainty, model (1) only specifies that we are interested in maximization of all objective functions f_i for $i \in I = \{1, 2, \ldots, m\}$. In order to make it operational, one needs to assume some solution concept specifying what it means to maximize multiple objective functions. The solution concepts are defined by aggregation functions $a : R^m \to R$. Thus the multiple criteria problem (1) is replaced with the (scalar) maximization problem

$$\max \{a(\mathbf{f}(\mathbf{x})) \ : \ \mathbf{x} \in \mathcal{F}\}$$

The most commonly used aggregation is based on the weighted mean where positive importance weights p_i $(i = 1, \ldots, m)$ are allocated to several scenarios

$$A_{\mathbf{p}}(\mathbf{y}) = \sum_{i=1}^{m} y_i p_i \tag{2}$$

The weights are typically normalized to the total 1 ($\sum_{i=1}^{m} p_i = 1$) with possible interpretation as scenarios (subjective) probabilities. The weighted mean allowing to define the importance of scenarios does not allow one to model the decision maker's preferences regarding distribution of outcomes. The latter is crucial when aggregating various realizations of the same (uncertain) outcome under several scenarios one needs to model risk averse preferences [7].

The preference weights can be effectively introduced within the fuzzy optimization methodology with the so-called Ordered Weighted Averaging (OWA) aggregation developed by Yager [15]. In the OWA aggregation the weights are assigned to the ordered values (i.e. to the smallest value, the second smallest and so on) rather than to the specific criteria. This guarantees a possibility to model various preferences with respect to the risk. Since its introduction, the OWA aggregation has been successfully applied to many fields of decision making [18,19,6]. The weighting of the ordered outcome values causes that the OWA optimization problem is nonlinear even for linear programming (LP) formulation of the original constraints and criteria. Yager [16] has shown that the OWA optimization can be converted into a mixed integer programming problem. We have shown [10] that the OWA optimization with monotonic weights can be formed as a standard linear program of higher dimension.

The OWA operator allows one to model various aggregation functions from the maximum through the arithmetic mean to the minimum. Thus, it enables modeling of various preferences from the optimistic to the pessimistic one. On the other hand, the OWA does not allow one to allocate any importance weights to specific scenarios. Actually, the weighted mean (2) cannot be expressed in terms of the OWA aggregations. Torra [12] has incorporated importance weighting into the OWA operator within the Weighted OWA (WOWA) aggregation introduced as a particular case of Choquet integral using a distorted probability as the measure. The WOWA average becomes the weighted mean in the case of equal all the preference weights and it is reduced to the standard OWA average for equal all the importance weights. Since its introduction, the WOWA operator has been successfully applied to many fields of decision making [14] including metadata aggregation problems [1].

In this paper we analyze solution procedures for optimization problems with the WOWA objective functions. A linear programming formulation is introduced for optimization of the WOWA objective with increasing preferential weights thus representing risk averse preferences. The paper is organized as follows. In the next section we introduce formally the WOWA operator and derive some alternative computational formula based on direct application of the preferential weights to the conditional means according to the importance weights. In Section 3 we introduce the LP formulations for maximization of the WOWA aggregation with increasing weights. Finally, in Section 4 we demonstrate computational efficiency of the introduced models.

2 The Weighted OWA Aggregation

Let $\mathbf{w} = (w_1, \ldots, w_m)$ be a weighting vector of dimension m such that $w_i \geq 0$ for $i = 1, \ldots, m$ and $\sum_{i=1}^{m} w_i = 1$. The corresponding OWA aggregation of outcomes $\mathbf{y} = (y_1, \ldots, y_m)$ can be mathematically formalized as follows [15]. First, we introduce the ordering map $\Theta : R^m \to R^m$ such that $\Theta(\mathbf{y}) = (\theta_1(\mathbf{y}), \theta_2(\mathbf{y}), \ldots, \theta_m(\mathbf{y}))$, where $\theta_1(\mathbf{y}) \geq \theta_2(\mathbf{y}) \geq \cdots \geq \theta_m(\mathbf{y})$ and there exists a permutation τ of set I such that $\theta_i(\mathbf{y}) = y_{\tau(i)}$ for $i = 1, \ldots, m$. Further, we apply the weighted sum aggregation to ordered achievement vectors $\Theta(\mathbf{y})$, i.e. the OWA aggregation has the following form:

$$A_{\mathbf{w}}(\mathbf{y}) = \sum_{i=1}^{m} w_i \theta_i(\mathbf{y}) \tag{3}$$

Yager [15] introduced a well appealing concept of the orness measure to characterize the OWA operators. $\text{orness}(\mathbf{w}) = \sum_{i=1}^{m} \frac{m-i}{m-1} w_i$. For the max aggregation representing the fuzzy 'or' operator with weights $\mathbf{w} = (1, 0, \ldots, 0)$ one gets $\text{orness}(\mathbf{w}) = 1$ while for the min aggregation representing the fuzzy 'and' operator with weights $\mathbf{w} = (0, \ldots, 0, 1)$ one has $\text{orness}(\mathbf{w}) = 0$. For the average (arithmetic mean) one gets $\text{orness}((1/m, 1/m, \ldots, 1/m)) = 1/2$. Actually, one may consider a complementary measure of andness defined as $\text{andness}(\mathbf{w}) = 1 - \text{orness}(\mathbf{w})$. OWA aggregations with orness smaller or equal $1/2$ are treated as and-like and they correspond to rather pessimistic (risk averse) preferences.

The OWA aggregations with increasing weights $w_1 \leq w_2 \leq \ldots \leq w_m$ define an and-like OWA operator. Actually, the andness properties of the OWA operators with increasing weights are total in the sense that they remain valid for any subaggregations defined by subsequences of their weights. Namely, for any $2 \leq k \leq m$ one gets $\sum_{j=1}^{k} \frac{k-j}{k-1} w_{i_j} \leq \frac{1}{2}$. Such total andness properties represent consequent risk averse preferences [7]. Therefore, we will refer to the OWA aggregation with increasing weights as the risk averse OWA.

Let $\mathbf{w} = (w_1, \ldots, w_m)$ be an m-dimensional vector of preferential weights such that $w_i \geq 0$ for $i = 1, \ldots, m$ and $\sum_{i=1}^{m} w_i = 1$. Further, let $\mathbf{p} = (p_1, \ldots, p_m)$ be an m-dimensional vector of importance weights such that $p_i \geq 0$ for $i = 1, \ldots, m$ and $\sum_{i=1}^{m} p_i = 1$. The corresponding Weighted OWA aggregation of outcomes $\mathbf{y} = (y_1, \ldots, y_m)$ is defined [12] as follows:

$$A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = \sum_{i=1}^{m} \omega_i \theta_i(\mathbf{y}) \quad \text{with} \quad \omega_i = w^*(\sum_{k \le i} p_{\tau(k)}) - w^*(\sum_{k<i} p_{\tau(k)}) \qquad (4)$$

where w^* is an increasing function interpolating points $(\frac{i}{m}, \sum_{k \le i} w_k)$ together with the point (0.0) and τ representing the ordering permutation for $\mathbf{y}$ (i.e. $y_{\tau(i)} = \theta_i(\mathbf{y})$). Moreover, function w^* is required to be a straight line when the point can be interpolated in this way. We will focus our analysis on the piecewise linear interpolation function w^* which is the simplest form of the required interpolation.

The WOWA aggregation covers the standard weighted mean (2) with weights p_i as a special case of equal preference weights ($w_i = 1/m$ for $i = 1, \ldots, m$). Actually, the WOWA operator is a particular case of Choquet integral using a distorted probability as the measure [3].

Example 1. Consider outcome vectors $\mathbf{y}' = (1, 3, 2, 4, 5)$ and $\mathbf{y}'' = (1, 1, 2, 6, 4)$ where individual outcomes correspond to five scenarios. While introducing preferential weights $\mathbf{w} = (0.05, 0.1, 0.15, 0.2, 0.5)$ one may calculate the OWA averages: $A_{\mathbf{w}}(\mathbf{y}') = 0.05 \cdot 5 + 0.1 \cdot 4 + 0.15 \cdot 3 + 0.2 \cdot 2 + 0.5 \cdot 1 = 2$ and $A_{\mathbf{w}}(\mathbf{y}'') = 0.05 \cdot 6 + 0.1 \cdot 4 + 0.15 \cdot 2 + 0.2 \cdot 1 + 0.5 \cdot 1 = 1.7$. Further, let us introduce importance weights $\mathbf{p} = (0.1, 0.1, 0.2, 0.5, 0.1)$ which means that results under the third scenario are 2 times more important then those under scenario 1, 2 or 5, while the results under scenario 4 are even 5 times more important. To take into account the importance weights in the WOWA aggregation (4) we introduce piecewise linear function

$$w^*(\xi) = \begin{cases} 0.05\xi/0.2 & \text{for } 0 \le \xi \le 0.2 \\ 0.05 + 0.10(\xi - 0.2)/0.2 & \text{for } 0.2 < \xi \le 0.4 \\ 0.15 + 0.15(\xi - 0.4)/0.2 & \text{for } 0.4 < \xi \le 0.6 \\ 0.3 + 0.2(\xi - 0.6)/0.2 & \text{for } 0.6 < \xi \le 0.8 \\ 0.5 + 0.5(\xi - 0.8)/0.2 & \text{for } 0.8 < \xi \le 1.0 \end{cases}$$

and calculate weights ω_i according to formula (4) as w^* increments corresponding to importance weights of the ordered outcomes, as illustrated in Fig. 1. In particular, one get $\omega_1 = w^*(p_5) = 0.025$ and $\omega_2 = w^*(p_5 + p_4) - w^*(p_5) = 0.275$ for vector $\mathbf{y}'$ while $\omega_1 = w^*(p_4) = 0.225$ and $\omega_2 = w^*(p_4 + p_5) - w^*(p_4) = 0.075$ for vector $\mathbf{y}''$. Finally, $A_{\mathbf{w},\mathbf{p}}(\mathbf{y}') = 0.025 \cdot 5 + 0.275 \cdot 4 + 0.1 \cdot 3 + 0.35 \cdot 2 + 0.25 \cdot 1 = 2.475$ and $A_{\mathbf{w},\mathbf{p}}(\mathbf{y}'') = 0.225 \cdot 6 + 0.075 \cdot 4 + 0.2 \cdot 2 + 0.25 \cdot 1 + 0.25 \cdot 1 = 2.55$.

Note that one may alternatively compute the WOWA values by using the importance weights to replicate corresponding scenarios and calculate then OWA aggregations. In the case of our importance weights $\mathbf{p}$ we need to consider five copies of scenario 4 and two copies of scenario 3 thus generating corresponding vectors $\tilde{\mathbf{y}}' = (1, 3, 2, 2, 4, 4, 4, 4, 4, 5)$ and $\tilde{\mathbf{y}}'' = (1, 1, 2, 2, 6, 6, 6, 6, 6, 4)$ of ten equally important outcomes. Original five preferential weights must be then applied respectively to the average of the two largest outcomes, the average of the next two largest outcomes etc. Indeed, we get $A_{\mathbf{w},\mathbf{p}}(\mathbf{y}') = 0.05 \cdot 4.5 + 0.1 \cdot 4 + 0.15 \cdot 4 + 0.2 \cdot 2.5 + 0.5 \cdot 1.5 = 2.475$ and $A_{\mathbf{w},\mathbf{p}}(\mathbf{y}'') = 0.05 \cdot 6 + 0.1 \cdot 6 + 0.15 \cdot 5 + 0.2 \cdot 2 + 0.5 \cdot 1 =$

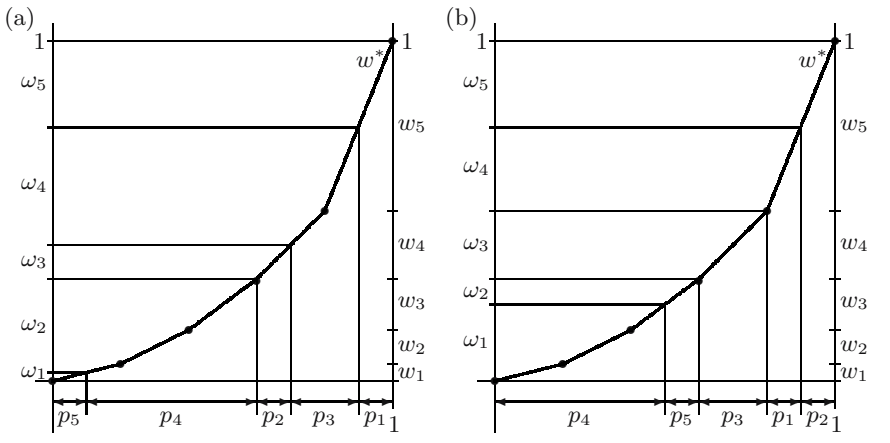


Fig. 1. Definition of weights ω_i for Example 1: (a) vector $\mathbf{y}'$, (b) vector $\mathbf{y}''$

2.55. We will further formalize this approach and take its advantages to build the LP computational models.

Function w^* can be defined by its generation function g with the formula $w^*(\alpha) = \int_0^\alpha g(\xi)\ d\xi$ Introducing breakpoints $\beta_i = \sum_{k \le i} p_{\tau(k)}$ and $\beta_0 = 0$ allows us to express $\omega_i = \int_0^{\beta_i} g(\xi)\ d\xi - \int_0^{\beta_{i-1}} g(\xi)\ d\xi = \int_{\beta_{i-1}}^{\beta_i} g(\xi)\ d\xi$ and the entire WOWA aggregation as

$$A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = \sum_{i=1}^{m} \theta_i(\mathbf{y}) \int_{\beta_{i-1}}^{\beta_i} g(\xi)\ d\xi = \int_0^1 g(\xi) \overline{F}_{\mathbf{y}}^{(-1)}(\xi)\ d\xi \tag{5}$$

where $\overline{F}_{\mathbf{y}}^{(-1)}$ is the stepwise function $\overline{F}_{\mathbf{y}}^{(-1)}(\xi) = \theta_i(\mathbf{y})$ for $\beta_{i-1} < \xi \le \beta_i$. It can also be mathematically formalized as follows. First, we introduce the right-continuous cumulative distribution function (cdf):

$$F_{\mathbf{y}}(d) = \sum_{i=1}^{m} p_i \delta_i(d) \quad \text{where} \quad \delta_i(d) = \begin{cases} 1 & \text{if } y_i \le d \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

which for any real (outcome) value d provides the measure of outcomes smaller or equal to d. Next, we introduce the quantile function $F_{\mathbf{y}}^{(-1)} = \inf\{\eta : F_{\mathbf{y}}(\eta) \ge \xi\}$ for $0 < \xi \le 1$ as the left-continuous inverse of the cumulative distribution function $F_{\mathbf{y}}$, and finally $\overline{F}_{\mathbf{y}}^{(-1)}(\xi) = F_{\mathbf{y}}^{(-1)}(1-\xi)$.

Formula (5) provides the most general expression of the WOWA aggregation allowing for expansion to continuous case. The original definition of WOWA allows one to build various interpolation functions w^* [13] thus to use different generation functions g in formula (5). We have focused our analysis on the the piecewise linear interpolation function w^*. Note, however, that the piecewise linear functions may be built with various number of breakpoints, not necessarily

m. Thus, any nonlinear function can be well approximated by an piecewise linear function with appropriate number of breakpoints. Therefore, we will consider weights vectors $\mathbf{w}$ of dimension n not necessarily equal to m. Any such piecewise linear interpolation function w^* can be expressed with the stepwise generation function $g(\xi) = nw_k$ for $(k-1)/n < \xi \le k/n$, $k = 1, \ldots, n$. This leads us to the following specification of formula (5):

$$A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = \sum_{k=1}^{n} w_k n \int_{(k-1)/n}^{k/n} \overline{F}_{\mathbf{y}}^{(-1)}(\xi)\, d\xi = \sum_{k=1}^{n} w_k n \int_{(k-1)/n}^{k/n} F_{\mathbf{y}}^{(-1)}(1-\xi)\, d\xi \quad (7)$$

Note that $n \int_{(k-1)/n}^{k/n} \overline{F}_{\mathbf{y}}^{(-1)}(\xi)\ d\xi$ represents the average within the k-th portion of $1/n$ largest outcomes, the corresponding conditional mean [9,11]. Hence, formula (7) defines WOWA aggregations with preferential weights $\mathbf{w}$ as the corresponding OWA aggregation but applied to the conditional means calculated according to the importance weights $\mathbf{p}$ instead of the original outcomes. Fig. 2 illustrates application of formula (7) for computation of the WOWA aggregations in Example 1.

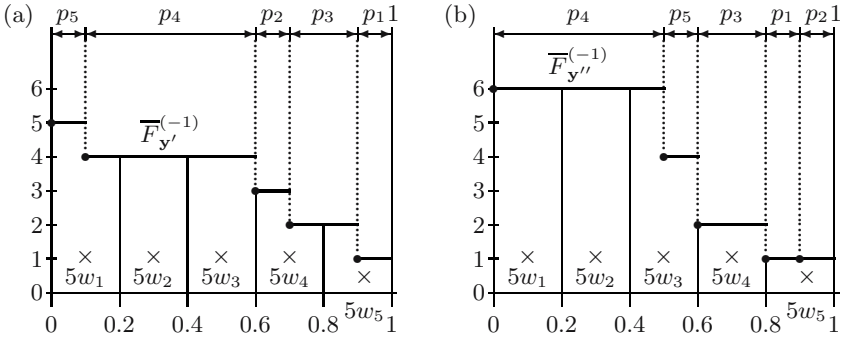


Fig. 2. Formula (7) applied to calculations in Example 1: (a) vector $\mathbf{y}'$, (b) vector $\mathbf{y}''$

We will treat formula (7) as a formal definition of the WOWA aggregation of m-dimensional outcomes $\mathbf{y}$ defined by m-dimensional importance weights $\mathbf{p}$ and n-dimensional preferential weights $\mathbf{w}$. We will focus our analysis on the WOWA aggregation defined by increasing weights $w_1 \le w_2 \le \ldots \le w_n$. Following formula (7), maximization of such WOWA aggregation models risk averse preferences since equally important unit of a smaller outcome is considered with a larger weight. This is mathematically represented by the convexity of function w^* as well as it may be viewed as andness of the WOWA operator [4] when considered as the OWA defined via the regular increasing monotone (RIM) quantifiers [17] ($\int_0^1 w^*(\xi)d\xi \le 0.5$).

3 The LP Model for WOWA Optimization

Formula (7) defines the WOWA value applying preferential weights w_i to importance weighted averages within quantile intervals. It may reformulated to use the tail averages

$$A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = \sum_{k=1}^{n} nw_k(L(\mathbf{y},\mathbf{p},1-\frac{k-1}{n}) - L(\mathbf{y},\mathbf{p},1-\frac{k}{n})) = \sum_{k=1}^{n} w'_k L(\mathbf{y},\mathbf{p},\frac{k}{n}) \quad (8)$$

where $L(\mathbf{y},\mathbf{p},\xi)$ is defined by left-tail integrating of $F_{\mathbf{y}}^{(-1)}$, i.e.

$$L(\mathbf{y},\mathbf{p},0) = 0 \quad \text{and} \quad L(\mathbf{y},\mathbf{p},\xi) = \int_0^{\xi} F_{\mathbf{y}}^{(-1)}(\alpha)d\alpha \quad \text{for } 0 < \xi \leq 1 \qquad (9)$$

while weights $w'_k = n(w_{n-k+1} - w_{n-k})$ for $k = 1,\ldots,n-1$ and $w'_n = nw_1$.

Graphs of functions $L(\mathbf{y},\mathbf{p},\xi)$ (with respect to ξ) take the form of convex piecewise linear curves, the so-called absolute Lorenz curves [8] connected to the relation of the second order stochastic dominance (SSD). Therefore, formula (8) relates the WOWA average to the SSD consistent risk measures based on the tail means [5] provided that the importance weights are treated as scenario probabilities.

Following (8), maximization of a risk averse WOWA aggregation defined by increasing weights $w_1 \leq w_2 \leq \ldots \leq w_n$

$$\max\{A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) \ : \ \mathbf{y} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \mathcal{F}\} \qquad (10)$$

results in problem

$$\max\{\sum_{k=1}^{n} w'_k L(\mathbf{y},\mathbf{p},\frac{k}{n}) \ : \ \mathbf{y} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \mathcal{F}\}$$

with positive weights w'_k.

According to (9), values of function $L(\mathbf{y},\mathbf{p},\xi)$ for any $0 \leq \xi \leq 1$ can be given by optimization:

$$L(\mathbf{y},\mathbf{p},\xi) = \min_{s_i} \{\sum_{i=1}^{m} y_i s_i : \sum_{i=1}^{m} s_i = \xi, \quad 0 \leq s_i \leq p_i \quad \forall\ i\ \} \qquad (11)$$

The above problem is an LP for a given outcome vector $\mathbf{y}$ while it becomes nonlinear for $\mathbf{y}$ being a vector of variables. This difficulty can be overcome by taking advantage of the LP dual to (11). Introducing dual variable t corresponding to the equation $\sum_{i=1}^{m} s_i = \xi$ and variables d_i corresponding to upper bounds on s_i one gets the following LP dual expression of $L(\mathbf{y},\mathbf{p},\xi)$

$$L(\mathbf{y},\mathbf{p},\xi) = \max_{t,d_i} \{\xi t - \sum_{i=1}^{m} p_i d_i : t - d_i \leq y_i,\ d_i \geq 0 \quad \forall\ i\} \qquad (12)$$

Therefore, maximization of the WOWA aggregation (10) can be expressed as follows

$$\begin{aligned} \max_{t_k,d_{ik},y_i,x_j} & \sum_{k=1}^{n} w'_k[\frac{k}{n}t_k - \sum_{i=1}^{m} p_i d_{ik}] \\ \text{s.t. } & t_k - d_{ik} \le y_i,\ d_{ik} \ge 0 \quad \forall\, i,k \\ & \mathbf{y} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \mathcal{F} \end{aligned}$$

Consider multiple criteria problems (1) with linear objective functions $f_i(\mathbf{x}) = \mathbf{c}_i\mathbf{x}$ and polyhedral feasible sets:

$$\max\{(y_1, y_2, \ldots, y_m)) \ :\ \mathbf{y} = \mathbf{C}\mathbf{x}, \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geqq \mathbf{0}\} \tag{13}$$

where $\mathbf{C}$ is an $m \times q$ matrix (consisting of rows $\mathbf{c}_i$), $\mathbf{A}$ is a given $r \times q$ matrix and $\mathbf{b} = (b_1, \ldots, b_r)^T$ is a given RHS vector. For such problems, we get the following LP formulation of the WOWA maximization (10):

$$\max_{t_k,d_{ik},y_i,x_j} \ \sum_{k=1}^{n} \frac{k}{n} w'_k t_k - \sum_{k=1}^{n}\sum_{i=1}^{m} w'_k p_i d_{ik} \tag{14}$$

$$\text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b} \tag{15}$$

$$\mathbf{y} - \mathbf{C}\mathbf{x} = \mathbf{0} \tag{16}$$

$$d_{ik} \ge t_k - y_i \quad \forall\, i,k \tag{17}$$

$$d_{ik} \ge 0 \quad \forall\, i,k;\ x_j \ge 0\ \forall\, j \tag{18}$$

Model (14)–(18) is an LP problem with $mn+m+n+q$ variables and $mn+m+r$ constraints. Thus, for problems with not too large number of scenarios (m) and preferential weights (n) it can be solved directly. Note that WOWA model (14)–(18) differs from the analogous deviational model for the OWA optimizations [10] only due to coefficients within the objective function (14) and the possibility of different values of m and n.

The number of constraints in problem (14)–(18) is similar to the number of variables. Nevertheless, for the simplex approach it may be better to deal with the dual of (14)–(18) than with the original problem. Note that variables d_{ik} in the primal are represented with singleton columns. Hence, the corresponding rows in the dual represent only simple upper bounds.

Introducing the dual variables: $\mathbf{u} = (u_l)_{l=1,\ldots,r}$, $\mathbf{v} = (v_i)_{i=1,\ldots,m}$ and $\mathbf{z} = (z_{ik})_{i=1,\ldots,m;\ k=1,\ldots,n}$ corresponding to the constraints (15), (16) and (17), respectively, we get the following dual:

$$\begin{aligned} \min_{z_{ik},v_i,u_l} \ & \mathbf{u}\mathbf{b} \\ \text{s.t.} \ & \mathbf{u}\mathbf{A} - \mathbf{v}\mathbf{C} \geqq \mathbf{0} \\ & v_i - \sum_{k=1}^{n} z_{ik} = 0 \quad \forall\, i \\ & \sum_{i=1}^{m} z_{ik} = \frac{k}{n} w'_k \quad \forall\, k \\ & 0 \le z_{ik} \le p_i w'_k \quad \forall\, i,k \end{aligned} \tag{19}$$

The dual problem (19) contains: $m+n+q$ structural constraints, $r+m$ unbounded variables and mn bounded variables. Since the average complexity of the simplex method depends on the number of constraints, the dual model (19) can be directly solved for quite large values of m and n. Moreover, the columns corresponding to mn variables z_{ik} form the transportation/assignment matrix thus allowing one to employ special techniques of the simplex SON algorithm [2] for implicit handling of these variables. Such techniques increase dramatically efficiency of the simplex method but they require a special tailored implementation. We have not tested this approach within our initial computational experiments based on the use of a general purpose LP code.

4 Computational Tests

In order to analyze the computational performances of the LP model for the WOWA optimization, similarly to [10], we have solved randomly generated problems of portfolio optimization according to the (discrete) scenario analysis approach [6]. There is given a set of securities for an investment $J = \{1, 2, \ldots, q\}$. We assume, as usual, that for each security $j \in J$ there is given a vector of data $(c_{ij})_{i=1,\ldots,m}$, where c_{ij} is the observed (or forecasted) rate of return of security j under scenario i (hereafter referred to as outcome). We consider discrete distributions of returns defined by the finite set $I = \{1, 2, \ldots, m\}$ of scenarios with the assumption that each scenario can be assigned the importance weight p_i that can be seen as the subjective probability of the scenario. The outcome data forms an $m \times q$ matrix $\mathbf{C} = (c_{ij})_{i=1,\ldots,m;j=1,\ldots,q}$ whose columns correspond to securities while rows $\mathbf{c}_i = (c_{ij})_{j=1,2,\ldots,q}$ correspond to outcomes. Further, let $\mathbf{x} = (x_j)_{j=1,2,\ldots,q}$ denote the vector of decision variables defining a portfolio. Each variable x_j expresses the portion of the capital invested in the corresponding security. Portfolio $\mathbf{x}$ generates outcomes

$$\mathbf{y} = \mathbf{C}\mathbf{x} = (\mathbf{c}_1\mathbf{x}, \mathbf{c}_2\mathbf{x}, \ldots, \mathbf{c}_m\mathbf{x})$$

The portfolio selection problem can be considered as an LP problem with m uniform objective functions $f_i(\mathbf{x}) = \mathbf{c}_i\mathbf{x} = \sum_{j=1}^{q} c_{ij}x_j$ to be maximized [6]:

$$\max \{\mathbf{C}\mathbf{x} \ : \ \sum_{j=1}^{q} x_j = 1, \quad x_j \geq 0 \quad \text{for } j = 1, \ldots, q\}$$

Hence, our portfolio optimization problem can be considered a special case of the multiple criteria problem and one may seek an optimal portfolio with some criteria aggregation. Note that the aggregation must take into account the importance of various scenarios thus allowing importance weights p_i to be assigned to several scenarios. Further the preferential weights w_k must be increasing to represent the risk averse preferences (more attention paid on improvement of smaller outcomes). Thus we get the WOWA maximization problem

$$\max \{A_{\mathbf{w},\mathbf{p}}(\mathbf{f}(\mathbf{x})) \ : \ \sum_{j=1}^{q} x_j = 1, \quad x_j \geq 0 \quad \text{for } j = 1, \ldots, q\} \tag{20}$$

Our computational tests were based on the randomly generated problems (20) with varying number q of securities (decision variables) and number m of scenarios. The generation procedure worked as follows. First, for each security j the maximum rate of return r_j was generated as a random number uniformly distributed in the interval $[0.05, 0.15]$. Next, this value was used to generate specific outcomes c_{ij} (the rate of return under scenarios i) as random variables uniformly distributed in the interval $[-0.75r_j, r_j]$. Further, strictly increasing and positive weights w_k were generated. The weights were not normalized which allowed us to define them by the corresponding increments $\delta_k = w_k - w_{k-1}$. The latter were generated as uniformly distributed random values in the range of 1.0 to 2.0, except from a few (5 on average) possibly larger increments ranged from 1.0 to $n/3$. Importance weights p_i were generated according to the exponential smoothing scheme, which assigns exponentially decreasing weights to older or subjectively less probable scenarios: $p_i = \alpha(1-\alpha)^{i-1}$ for $i = 1, 2, \ldots, m$ and the parameter α is chosen for each test problem size separately to keep the value of p_m around 0.001.

We tested solution times for different size parameters m and q. The basic tests were performed for the standard WOWA model with $n = m$. However, we also analyzed the case of larger n for more detailed preferences modeling, as well as the case of smaller n thus representing a rough preferences model. For each number of decision variables (securities) q and number of criteria (scenarios) m we solved 10 randomly generated problems (20). All computations were performed on a PC with the Pentium 1.7GHz processor employing the CPLEX 9.1 package. The 120 seconds time limit was used in all the computations.

Table 1. Solution times [s] for the primal model (14)–(18)

Number of scenarios (m)	Number of variables (q)							
	10	20	50	100	150	200	300	400
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.1	0.1	0.1	0.1	0.1
50	1.8	2.5	3.5	4.0	4.1	4.0	3.9	4.0
100	55.7	77.4	89.5	^{2}106.3	^{7}117.7	–	–	–

In Tables 1 and 2 we show the solution times for the primal (14)–(18) and the dual (19) forms of the computational model, being the averages of 10 randomly generated problems. Upper index in front of the time value indicates the number of tests among 10 that exceeded the time limit. The empty cell (minus sign) shows that this occurred for all 10 instances. Both forms were solved by the CPLEX code without taking advantages of the constraints structure specificity. The dual form of the model performs much better in each tested problem size. It behaves very well with increasing number of variables if the number of scenarios does not exceed 100. Similarly, the model performs very well with increasing number of scenarios if only the number of variables does not exceed 20.

Table 2. Solution times [s] for the dual model (19)

Number of scenarios (m)	Number of variables (q)							
	10	20	50	100	150	200	300	400
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.1
50	0.1	0.1	0.4	0.7	0.9	1.0	1.5	1.9
100	0.7	1.0	3.4	19.5	24.9	30.0	33.6	38.9
150	2.3	3.5	7.9	80.2	–	–	–	–
200	5.6	7.9	17.1	–	–	–	–	–
300	20.0	30.6	[1]89.3	–	–	–	–	–
400	51.9	92.8	–	–	–	–	–	–

Table 3. Solution times [s] for different numbers of preferential weights ($m = 100$, $q = 50$)

Number of preferential weights (n)								
3	5	10	20	50	100	150	200	300
0.0	0.1	0.1	0.4	3.3	3.4	2.6	3.6	6.5

Table 3 presents solution times for different numbers of the preferential weights for problems with 100 scenarios and 50 variables. It can be noticed that the computational efficiency can be improved by reducing the number of preferential weights which can be reasonable in non-automated decision making support systems and actually provides very good results for portfolio optimization problems [5]. On the other hand increasing the number of preferential weights and thus the number of breakpoints in the interpolation function does not induce the massive increase in the computational complexity.

5 Concluding Remarks

The problem of averaging outcomes under several scenarios to form overall objective functions is of considerable importance in decision support under uncertainty. The WOWA aggregation [12] represents such a universal tool allowing one to take into account both the preferential weights allocated to ordered outcomes and the importance weights allocated to several scenarios. The ordering operator used to define the WOWA aggregation is, in general, hard to implement. We have shown that the WOWA aggregations with the increasing weights can be modeled by introducing auxiliary linear constraints. Hence, an LP problem with the risk averse WOWA aggregation can be formed as a standard linear program and it can be further simplified by taking advantages of the LP duality.

Initial computational experiments show that the formulation enables to solve effectively medium size problems. Actually, the number of 100 scenarios covered by the dual approach to the LP model seems to be quite enough for most applications, including the fuzzy aggregations and decisions under risk. The problems

have been solved directly by general purpose LP code. Taking advantages of the constraints structure specificity may remarkably extend the solution capabilities. In particular, the simplex SON algorithm [2] may be used for exploiting the LP embedded network structure in the dual form of the model.

References

1. Damiani, E., De Capitani di Vimercati, S., Samarati, P., Viviani, M.: A WOWA-based aggregation technique on trust values connected to metadata. Electronic Notes Theor. Comp. Sci. 157, 131–142 (2006)
2. Glover, F., Klingman, D.: The simplex SON method for LP/embedded network problems. Math. Progr. Study 15, 148–176 (1981)
3. Grabisch, M., Orlovski, S.A., Yager, R.R.: Fuzzy aggregation of numerical preferences. In: Fuzzy sets in decision analysis, operations research and statistics, pp. 31–68. Kluwer, Dordrecht (1999)
4. Liu, X.: Some properties of the weighted OWA operator. Man and Cyber. B 368, 118–127 (2006)
5. Mansini, R., Ogryczak, W., Speranza, M.G.: Conditional Value at Risk and Related Linear Programming Models for Portfolio Optimization. Annals of Oper. Res. 152, 227–256 (2007)
6. Ogryczak, W.: Multiple criteria linear programming model for portfolio selection. Annals Oper. Res. 97, 143–162 (2000)
7. Ogryczak, W.: Multiple criteria optimization and decisions under risk. Control & Cyber. 31, 975–1003 (2002)
8. Ogryczak, W., Ruszczyński, A.: Dual stochastic dominance and related mean-risk models. SIAM J. Optimization 13, 60–78 (2002)
9. Ogryczak, W., Śliwiński, T.: On equitable approaches to resource allocation problems: the conditional minimax solution. J. Telecom. Info. Tech. 3/02, 40–48 (2002)
10. Ogryczak, W., Śliwiński, T.: On solving linear programs with the ordered weighted averaging objective. Eur. J. Opnl. Res. 148, 80–91 (2003)
11. Ogryczak, W., Zawadzki, M.: Conditional Median — A Parametric Solution Concept for Location Problems. Annals of Oper. Res. 110, 167–181 (2002)
12. Torra, V.: The weighted OWA operator. Int. J. Intell. Syst. 12, 153–166 (1997)
13. Torra, V.: The WOWA operator and the interpolation function W*: Chen and Otto's interpolation method revisited. Fuzzy Sets Syst. 113, 389–396 (2000)
14. Valls, A., Torra, V.: Using classification as an aggregation tool for MCDM. Fuzzy Sets Syst. 115, 159–168 (2000)
15. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Trans. Systems, Man and Cyber. 18, 183–190 (1988)
16. Yager, R.R.: Constrained OWA aggregation. Fuzzy Sets Syst. 81, 89–101 (1996)
17. Yager, R.R.: Quantifier guided aggregation using OWA operators. Int. J. Intell. Syst. 11, 49–73 (1996)
18. Yager, R.R., Filev, D.P.: Essentials of Fuzzy Modeling and Control. Wiley, New York (1994)
19. Yager, R.R., Kacprzyk, J.: The Ordered Weighted Averaging Operators: Theory and Applications. Kluwer, Dordrecht (1997)

Transposing the Sociology of Organized Action into a Fuzzy Environment

Sandra Sandri[1] and Christophe Sibertin-Blanc[2]

[1] IIIA/CSIC, Artificial Intelligence Research Institute / Spanish National Research Council
Campus UAB, 08193 Bellaterra, Catalonia, Spain
sandri@iiia.csic.es
[2] IRIT, Université Toulouse 1, 21 Allées de Brienne, 31042 Toulouse Cedex, France
sibertin@univ-tlse1.fr

Abstract. In this work, we address the transposition of a fragment of the modeling of the Sociology of Organized Action to the fuzzy setting. We present two different ways of developing fuzzy models in this context, that depend on the kind of available data furnished by the user: one based on the extension principle and another using fuzzy rule-based inference with similarity relations. We illustrate our approach with an example from the sociology literature.

1 Introduction

Economics is mainly concerned with the creation of wealth, and games in this context help to understand how economic actors can accumulate gains in a competitive environment, each actor following a methodological individualistic strategy [1]. On the other hand, sociology is mainly concerned with the kinds of relationships that make human beings build the societies in which they live, and social games study the relationships that social actors can produce, resulting in a state of affairs in which each actor accepts both his own position and the position of others.

Here we are interested in social games, inspired by a sociology theory called the *Sociology of Organized Action* (SOA), or *Strategic Analysis*, initiated by M. Crozier [2] and notably further developed by E. Friedberg, (see [3]). Crozier and Friedberg part from the notion of limited rationality due to March and Simon and extract practical consequences from it [6].

SOA addresses social organisations or, more generally, *Systems of Concrete Actions* (SAC), that interact with an environment, pursue some goals, and manage means and resources that are used by the members of the organisation according to some rules.

A SAC is composed of "numerous differentiated actors interacting in a non-trivial way among each other" [3], and it defines an interaction context which structures and motivates the cooperation among social actors. Any organization features regulation phenomena that ensure its relative stability and the balance of social relationships (as long as there is no change in the goals and means). This regulation is enacted by the organization members, and SOA intends to explain how and why social actors behave as they do.

According to SOA, the behaviour of a member of an organisation is fully explained neither by the formal and informal rules, norms, etc. of the organisation, nor by each

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 791–802, 2007.

member individual particularities resulting from his history, nature, etc. Social actors have a strategic behaviour, i.e., they perform actions with the intention to achieve some goals, and each actor aims, as a meta-objective, at having enough power to preserve or increase his autonomy and capacity of action. This power results from and is exerted through the mastering of uncertainty zones (UZ), which are the resources that are needed by other actors for their actions. The actor (or group of actors) that masters/controls an uncertainty zone sets its exchange rules, i.e., how well other actors can access and use this particular resource.

UZs are the means of the power relationships between social actors, and a balance results from the fact that each actor both controls some UZs and depends on some others. Moreover, SOA assumes that each actor behaves strategically although he has only bounded rationality capabilities [11].

SOA has been formalized in [10], and is employed in the interactive environment SocLab (available at sourceforge.net), which allows the user to edit the structure of a SAC and to simulate the behaviour of the social actors [8].

Human knowledge is usually imperfect, tainted with imprecision, vagueness and uncertainty. That is specially true in what regards complex objects such as social organizations. To model and manipulate such knowledge computationally, it is important to address these issues in order to lose as little of the original information as possible. Fuzzy sets offers a very human-friendly means of modeling knowledge, and the operators provided by fuzzy sets theory are able to capture many of the ways human beings manipulate knowledge. In this work, we address the transposition of the modeling of SOA by [10] to the fuzzy setting using two different approaches, that depend on the kind of available data furnished by the user: one is based on the extension principle and the other on fuzzy rule-based inference with similarity relations.

This work is organized as follows. In Section 2, we present an example from the sociology literature that will be used to illustrate the various issues addressed in the remaining of the text. In Section 3, we present a fragment the modeling of SOA in [10] and then in Section 4 we present our two fuzzy approaches to deal with this fragment. Section 5 finally brings the conclusion.

2 Running Example

To illustrate how SOA analyses a system of concrete action and how we formalize this analysis, let us consider a classical example from strategic analysis [12].

Travel-tours is a tour operator having two agencies, TRO1 and TRO2, both of them situated in Trouville. These last months, the results of the TRO1 agency have increased, whereas the ones of TRO2 agency remained stable, or even decreased. The regional director decides to reward the TRO1 agency for its merits. He proposes then to regularize the situation of Agnès, a secretary of the company and to affect her exclusively to TRO1. She has been temporary employed by Travel-tours for several months, and even if she is formally attached to TRO1, she works half time in each agency and is obliged to move between the two jobs. Both Agnès and the TRO1 agency's director, Paul, should be glad with this proposal: Agnès would have a permanent job contract and would be

relieved to split her work in two parts, whereas Paul would have a full-time secretary at his disposal in TRO1.

However, both of them vigorously refuse the proposal. How should we understand this situation ? An strategic analysis, by identifying the uncertainty zones, shows that both Paul and Agnès are rationally right to be opposed to this organizational change, because it would decrease their respective power.

Indeed, a more attentive analysis of the case reveals that TRO2 is more inventive than TRO1 in designing travel packages, while TRO1 includes a very efficient commercial staff; being aware of the TRO2 agency's activity, the secretary provides information to the director so that TRO1 takes full advantage of finalizing TRO2's ideas. On the other hand, for personal reasons, to get a steady job is not one of Agnès' short-time objectives. Moreover, she greatly appreciates that none of the TRO1 and TRO2 directors has the possibility to exert a precise control on her work.

Thus the situation shift would increase the control of the director on the secretary's activities (something she does not want), and the director would lose the information given by the secretary on TRO2 (something he does not want).

3 Formalization of a Fragment of SOA

The basic social game formalization, as given in [10], can be described by the 5-tuple $G =< A, R, effect, m, stake, payoff >$, where:

- $A = \{a_1, ..., a_N\}$ is a set of social actors.
- $R = \{r_1, ..., r_M\}$ is a set of resources, each of which needed by one or more actors in A and controlled by one actor in A; the state of a resource r_i at a given moment is denoted by s_i, and modeled by a value in the interval $[-1, 1]$. The overall state of the game is defined by the state of all the resources, described by a vector $s = (s_1, s_2, ..., s_M) \in [-1, 1]^M$.
- $m : R \rightarrow A$ indicates which actor controls which resource; it is always assumed that each actor controls at least one resource, and thus $M \geq N$. The access of each resource r_i by its controller is bounded by its minimum and maximum values $b_i = [b\_min_i, b\_max_i] \subseteq [-1, 1]$. In the present implementation, a resource is controlled by a single actor. Also, the bounds on the access to a resource are constant values.
- $effect_i : A \times [-1, 1] \rightarrow [-10, 10]$ is a function that models how well an actor a can access resource r_i in its current state s_i. The worst and best possible accesses are respectively modeled by -10 and +10.
- $stake : A \times R \rightarrow [0, 10]$ is a function that expresses how important it is for an actor to access a resource. Each actor distributes the same number of stake points to the resources: $\forall a \in A, \sum_{r_i \in R} stake(a, r_i) = 10$. For a resource $r_k \in R$, $stake(a, r_k) = 0$ means that a has no need for r_k, whereas $stake(a, r_k) = 10$ means that r_k is the unique resource needed by a.
- $payoff : A \times [-1, 1]^M \rightarrow \mathbb{R}$ is a function that expresses how much an actor is comfortable with a state s of the game. A high payoff corresponds to a state where the actor has a good quality of access to the resources that are important to him.

At the moment, the implementation in SocLab of function $payoff$ is restricted to

$$payoff(a, s) = \sum_{r_i \in R} stake(a, r_i) * effect_i(a, s_i),$$

but other functions can be envisaged, such as non-linear combination functions, effect functions regarding two or more resources conjointly, payoff functions in which stakes do not play an explicit role, etc.

Note that $stakes$ and $effect$ are relevant in this framework, independently of function $payoff$ using them or not, because many important concepts of SOA, as modeled in [10], depend on such items. For instance, the $relevance$ of a resource and the $autonomy/subordination$ of an actor depend on the stakes:

- $relevance(r) = \sum_{a \in A} stake(a, r)$
- $autonomy(a) = \sum_{r_i \in R:m(r)=a} stake(a, r)$
- $subordination(a) = \sum_{r_i \in R:m(r)\neq a} stake(a, r)$

Functions autonomy and subordination are bounded in the interval $(0, 10]$ and $relevance$ is bounded in $(0, 10k]$ where $k =\mid A \mid$. Also, $autonomy(a) + subordination(a) = 10$.

In the formalization adopted here, some important concepts that depend on function $effect$ are for instance the force of a relation upon an actor, the relation of power between actors, the consequent dependencies of an actor upon another, etc. Yet other concepts, related to the accessibility of the resources, depend on the size of the interval $[b\_min_i, b\_max_i]$.

3.1 Playing a Social Game

The actor that controls a given resource is the one who decides the state of that resource. An action of actor a is a vector of the form $(m_i)_{r_i \in m^{-1}(a)}$, where m_i is the move to be applied to s_i (the current state of r_i), and that move is feasible if $s_i + m_i \in [b\_min, b\_max]$. A step of the game occurs when each actor has chosen a feasible move m_i for each resource r_i that he controls, and the game goes from state $(s_1, s_2, ..., s_M)$ to state $(s_1\acute{}, s_2\acute{}, ..., s_M\acute{})$, where $s_i\acute{} = s_i + m_i$.

The game is repeated until it becomes stabilized, or stationary: each actor plays the null action and no longer changes the state of the resources he controls. Such a state of the game is considered a social equilibrium, a balanced situation that is satisfying and accepted by all of the actors of the game. In most human organisations, social games are positive sum games: each actor gets some profit from being cooperative with each other, because others will also be cooperative in return. Thus, typical *social equilibria* are Pareto maxima: each actor has a high satisfaction, and any increase of it would entail a decrease of the satisfaction of another actor, and thus produce a situation that would not be accepted by all the actors.

3.2 Implementation of the Travel-Tours Example

The formalisation of this case in [10] includes two actors $A = \{Director, Secretary\}$, and three resources $R = \{r_1, r_2, r_3\}$, where r_1 stands for the stability of the secretary's

job, r_2 refers to the content of the secretary's work, and r_3 that represents the information about the activities of TRO2 agency. The director masters resources r_1 and r_2 and the secretary masters r_3; therefore, the values of s_1 and s_2 are set by the director and that of s_3 is set by the secretary.

Table 1 presents the stakes that each actor place on the resources and the effects of the resources' states upon the actors (we also indicate which actor controls each of the resources). For instance, concerning the information about TRO2 resource, the more the secretary gives information, the best it is for the director, who uses this information to improve the activity of TRO1 agency, but the worse it can be for herself if someone from TRO2 discovers that she furnishes information to TRO1. The director puts a high stake on this resource (as far as the social game is restricted to his relations with the secretary) because bringing this information is the most important contribution of the secretary to the agency, whereas giving or not the information does not have a high effect on her. Similar considerations explain the values concerning the stability of the job and control of work resources.

Table 1. Stakes and effects in the Travel-Tours case study (taken from [10])

stakes	Director	Secretary
r_1 (D)	1	2
r_2 (D)	2	7
r_3 (S)	7	1

effects	Director	Secretary
r_1 (D)	$3s_1$	$10s_1$
r_2 (D)	$-3(s_2)^2$	$7s_2$
r_3 (S)	$10s_3$	$-2 \mid s_3 \mid$

From Table 1 we obtain $relevance(r_1)=3, relevance(r_2)=9$ and $relevance(r_3)=8$, $autonomy(Director) = 3$ ($subordination(Director) = 7$) and $autonomy(Secretary) = 1$ ($subordination(Secretary) = 9$). We can directly state the payoff of each actor:

- $payoff(Director) = 3s_1 - 6(s_2)^2 + 70s_3$
- $payoff(Secretary) = 20s_1 + 49s_2 - 2 \mid s_3 \mid$

The boundaries for the resources are given as $b_1 = [-.4, .4], b_2 = [-.3, .7] and b_3 = [-.3, .8]$. Some interesting states are

- the secretary's optimum: $s_a == (.4, .7, 0)$ with $payoff(Secretary) = 42.3$ and $payoff(Director) = -1.7$.
- the director's optimum: $s_b = (.4, 0, .8)$ with $payoff(Director) = 57.2$ and $payoff(Secretary) = 6.4$.
- a Pareto optimum: $s_c = (.4, .7, .8)$, with $payoff(Director) = 54.3$ and $payoff(Secretary) = 40.7$.

4 A Fuzzy Modelization of SOA

The interest to allow fuzzyness in SOA is due in part to the fact that humans furnish information, and perceive the information presented to them, imperfectly. In the stage that SOA is insofar mathematically modeled by [10], fuzzy formalisms can be used

to model vagueness in the definition of values of stakes and moves in the sates, of functions such as $effect$ and $payoff$, on the boundaries of the resources accessibility and on functions depending on these items.

In the following, we present some basic definitions from the fuzzy systems literature that are needed in the rest of the paper. We then present a fuzzy modeling to the problem in two approaches whose choice depend on the type of available data. We illustrate both approaches using the Travel-Tours example.

4.1 Basic Definitions and Notations

In this section we recall some basic definitions that are used in the rest of the paper and provide some notation. Most of the definitions and remarks are well-known in the literature.

In the rest of the paper, unless stated otherwise, we shall work with fuzzy subsets of the real line, so the domain U below is assumed to be $\mathbb{R}$. The core (respec. support) of a fuzzy set $A : U \rightarrow [0,1]$ is defined as $core(A) = \{x \mid A(x) = 1\}$ (respec. $supp(A) = \{x \mid A(x) > 0\}$). For any $\alpha \in [0,1]$, the α-cut of A is defined as $[A]_\alpha = \{x \in U \mid A(x) \geq \alpha\}$. A is said to be *normalized* when there exists x such that $A(x) = 1$, and *convex* when for all x, y, z, if $x \leq y \leq z$, $A(y) \geq \min(A(x), A(z))$. A linear by parts convex fuzzy set A, a trapezoid, is denoted as $< a_1, a_2, a_3, a_4 >$ where $supp(A) = (a_1, a_4)$ and $core(A) = [a_2, a_3]$. When A is triangular, i.e. $a_2 = a_3$, the notation is simplified to $< a_1, a_2, a_4 >$. A normalized convex fuzzy set is called a *fuzzy interval*.

Given any function f from X to Y, the *extension principle* permits us to extend f to fuzzy sets. That is,

$$\hat{f}(A)(y) = \max_{x \in X, f(x)=y} A(x)$$

This expression is extended easily into functions $f : X_1, \ldots, X_N \rightarrow Y$. For illustration, when f is the sum, the extension principle permits us to compute the fuzzy sum of fuzzy sets A and B:

$$C_{A \oplus B}(y) = \sup_{(x_1,x_2)/x_1+x_2=y} \min(A(x_1), B(x_2)).$$

Operations on fuzzy sets are greatly simplified when their membership functions can be expressed by means of two functions $L, R : \mathbb{R}^+ \rightarrow [0,1]$, called *shape functions*, and L (and likewise R) is such that (a) $L(0) = 1$, (b) $\forall u > 0, L(u) < 1$, (c) $\forall u < 1, L(u) > 0$, and (d) $L(1) = 0$ or $[L(u) > 0, \forall u$ and $L(\infty) = 0]$. A fuzzy interval A is said to be of the LR type when its membership function can be defined with shape functions L and R and four parameters $(\underline{m}, \overline{m}) \in \mathbb{R}^2$, α and β as [4]

$$A(x) = \begin{cases} L(\frac{\underline{m}-x}{\alpha}) & \text{if } x \leq \underline{m} \\ 1, & \text{if } \underline{m} \leq x \leq \overline{m} \\ R(\frac{x-\overline{m}}{\beta}), & \text{if } x \leq \overline{m} \end{cases}$$

and is denoted by $(\underline{m}, \overline{m}, \alpha, \beta)$. The trapezoid fuzzy interval $(\underline{m}, \overline{m}, \alpha, \beta)$ is written in our notation as $< \underline{m} - \alpha, \underline{m}, \overline{m}, \overline{m} + \beta >$.

Let $A =< a_1, a_2, a_3, a_4 >$ and $B =< b_1, b_2, b_3, b_4 >$ be two LR intervals and δ be a (precise) constant in $\mathbb{R}$. Using the properties of LR intervals, the fuzzy sum, fuzzy subtraction, and the product by a precise constant are defined, in our notation, as

- $A \oplus B =< a_1 + b_1, a_2 + b_2, a_3 + b_3, a_4 + b_4 >$
- $A \ominus B =< a_1 - b_4, a_2 - b_3, a_3 - b_2, a_4 - b_1 >$
- $A \otimes \delta =< \delta a_1, \delta a_2, \delta a_3, \delta a_4 >$

The fuzzy product of two LR intervals (the same for the division) is not necessarily LR and to express it as LR one has to use some of the approximations proposed in the literature [4].

4.2 Fuzzy Framework Using the Extension Principle

In the first fuzzy approach, constant values such as stakes, moves or alternatively new values for resources states and boundaries, are simply allowed to be modeled by fuzzy sets. The biggest concern in this case regards the restrictions that the SOA model used here imposes on these values. In the following we analyze these issues.

- **States**
  A state s_i in the non-fuzzy framework varies in interval $[-1, 1]$. We denote a *fuzzy state* by s_i^*, with membership function $s_i^* : [-1, 1] \rightarrow [0, 1]$.
- **Stakes**
  A stake $stake(a, r)$ in the non-fuzzy framework varies in interval $[0, 10]$. We denote a *fuzzy stake* by $stake^*(a, r)$, with membership function $stake^*(a, r) : [0, 10] \rightarrow [0, 1]$. The stakes add up to 10 for each actor a, so for the case of fuzzy stakes we impose the restriction $10 \in core(F)$, where $F = stake^*(a, r_1) \oplus ... \oplus stake^*(a, r_M)$.
- **Boundary values**
  The values bounding the accessibility of a resource by its own controlling actor, $[b\_min_i, b\_max_i]$, vary in interval $[-1, 1]$ in the non-fuzzy framework. We denote a *lower fuzzy boundary* by $b\_min_i^*$, with membership function $b\_min_i^* : [-1, 1] \rightarrow [0, 1]$ and an *upper fuzzy boundary* by $b\_max_i^*$, with membership function $b\_max_i^* : [-1, 1] \rightarrow [0, 1]$. Let $b\_min_i^* =< a_1, b_1, c_1, d_1 >$ and $b\_max_i^* =< a_2, b_2, c_2, d_2 >$. Since $b\_min_i \leq b\_max_i$, in the case of fuzzy boundaries we impose the restrictions (i) $b_1 \leq a_2$, (ii) $c_1 \leq b_2$ and (iii) $d_1 \leq c_2$.

Functions that use these constants are extended in this fuzzy framework using the extension principle. For example, functions $relevance$, $autonomy$ and $subordination$ are easily extended substituting the sum operator by its fuzzy counterpart. Linear $effect$ functions are likewise easily extended and all these functions can be implemented straightforwardly when the membership functions are LR [4]. However, non-linear functions (e.g. $effect_2(Director, .)$) having LR fuzzy sets as arguments do not necessarily result in LR fuzzy sets. In this case, to get an efficient computation the best alternative is to approximate the result by a LR fuzzy set. The same problem occurs in the computation of the payoff function when both the $effect$ functions and the stakes are fuzzy. Several approximations are proposed in the literature for the product of fuzzy sets [4]; special care would be required for more complex functions. Fortunately, the SOA framework seems to need only reasonably simple functions for an acceptable formalization.

For example, let us suppose we have fuzzy stakes 1^*, 2^* and 7^* as illustrated in Figure 1. The payoff functions for the director and secretary described in 3.2 are given in the fuzzy framework as

- $payoff(Director) = 1^* \otimes (3 \otimes s_1^*) \oplus 2^* \otimes (-3 \otimes (s_2^*)^2) \oplus 7^* \otimes (10 \otimes s_3^*) = < 0,3,6 > \otimes s_1^* \ominus < 3,6,9 > \otimes (s_2^*)^2 \oplus < 60,70,80 > \otimes s_3^*$
- $payoff(Secretary) = 2^* \otimes (10 \otimes s_1^*) \oplus 7^* \otimes (7 \otimes s_2^*) \oplus 1^* \otimes (-2 \otimes \mid s_3^* \mid) = < 10,20,30 > \otimes s_1^* \oplus < 42,49,56 > \otimes s_2^* \ominus < 0,2,4 > \otimes \mid s_3^* \mid$

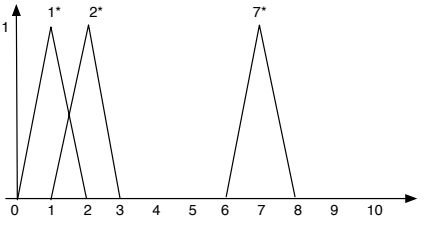


Fig. 1. Fuzzy stakes

Considering fuzzy stakes but non-fuzzy states, the payoff functions are reduced to

- $payoff(Director) = < x,y,z >$, where $x = 60s_3 - 9(s_2)^2$, $y = 3s_1 + 70s_3 - 6(s_2)^2$, $z = 6s_1 + 80s_3 - 3(s_2)^2$.
- $payoff(Secretary) = < o,p,q >$, where $o = 10s_1 + 42(s_2)^2 - 4|s_3|$, $p = 20s_1 + 49(s_2)^2 - 2|s_3|$, $q = 30s_1 + 56(s_2)^2$.

Results to some specific states are given in Table 2 below. Note that since the core of the fuzzy stakes correspond to the precise values given in the crisp case, the core of payoff results consequently correspond to the payoff values in the crisp case.

Table 2. Payoffs in the Travel-Tours example using the extension principle

state	Director	Secretary
$s_a = (.4,.7,0)$	<-4.4, -1.7,.9>	<33.4, 42.3, 51.2>
$s_b = (.4,0,.8)$	<48, 57.2,66.4>	<.8,6.4,12>
$s_c = (.4,.7,.8)$	<43.6, 54.3,64.9>	<30.2,40.7,51.2>

A similar treatment is employed when we have fuzzy states and non-fuzzy stakes. For instance, if instead of state $(.4,.7,0)$ we had state $(.4,.7^*,0)$ with $.7^* =< .6,.7,.8 >$, we would obtain $< 37.4,42.3,47.2 >$ as the secretary's payoff. However, the use of functions such as $(s_2)^2$ would make it necessary to use an approximation if we wanted to keep on using LR fuzzy sets; for example $(.4,.7^*,0)$ is linear whereas $(.4,.7^*,0) \otimes (.4,.7^*,0)$ is not so. The same problem would happen when we have fuzzy stakes and fuzzy states in the same function as explained above.

4.3 Rule-Based Fuzzy Framework

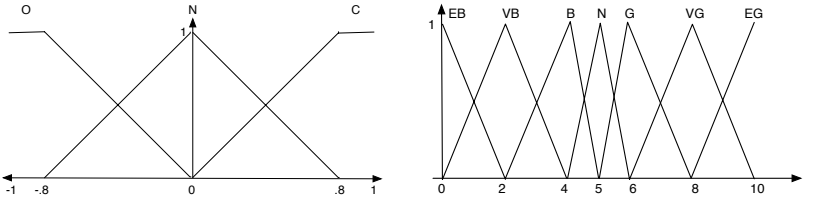
Even though the expressions used to model the effect and payoff functions may be mathematically simple, such as those employed in our example, it may not be so simple for someone modeling a social organization to create them from scratch. When that is the case, we propose to model such functions using a fuzzy rule-based approach. This can be done using the framework proposed in [9] for the general case of multicriteria decision making. Instead of describing the original method formally and then show its application to the running example, we will explain the method as we develop the example in the fuzzy framework.

First of all, now the payoff of an actor in our example will be given by a trade-off between the effect of the state on the resources he controls ($effect_{aut}$) and the effect on the resources he depends on but does not control ($effect_{sub}$):

$$payoff(a,s) = autonomy(a) \otimes effect_{aut}(a,s) \oplus subordination(a) \otimes effect_{sub}(a,s),$$

where s is the state vector. We propose to obtain the values for $effect_{aut}$ and $effect_{sub}$ functions through the use of fuzzy rule bases.

Let us consider a simple rule base implementation. Here, each resource can be in one of three following states - opposing, neutral or cooperative, denoted by O, N and C - according to the orientation of the control by the actor mastering that resource. As for the effect of the state of a resource upon an actor of the game, it can be extremely bad (EB), very bad (VB), bad (B), null (N), good (G), very good (VG) or extremely good (EG). In each of the rule bases, the input variables refer to the states of resources, whereas the output one (either $effect_{aut}$ or $effect_{sub}$) refers to the effect of the state of those resources on an actor. The fuzzy terms relative to the input and output linguistic terms used in the example are given in Figure 2. In Tables 3 and 4 we bring the rule bases relative to the director and the secretary respectively.



a) b)

Fig. 2. Fuzzy terms for the $effect$ rule bases: a) input and b) output variables

In order to infer the values of $effect_{aut}$ and $effect_{sub}$ from the rule bases, we propose to use the mechanism presented in [7] that employs gradual rules with similarity relations. The knowledge base is interpreted as a set of *gradual rules*, in the sense of [5], using residuated implication operators to perform the inference. Thus, a rule "If x is A_i then y is B_i" induces a fuzzy relation between input and output values which is defined as $R_i(x,y) = (A_i \rightarrow_\top B_i)(x,y) = A_i(x) \rightarrow_\top B_i(y)$, where $\rightarrow_\top$ is a

Table 3. Rule bases for the director

$effect_{aut}$

$s_2 \mid s_1$	O	N	C
C	B	B	B
N	B	N	G
O	B	B	B

$effect_{sub}$

s_3	O	N	C
	EB	N	EG

Table 4. Rule bases for the secretary

$effect_{sub}$

$s_2 \mid s_1$	O	N	C
C	G	VG	EG
N	B	N	G
O	EB	VB	B

$effect_{aut}$

s_3	O	N	C
	B	N	B

residuated implication operator. The output of a a set of gradual rules K in the case of precise input value x_0, as in our problem, is given by $output(K, \{x_0\}) = R_K(x_0, y) = \min_{i \in I} B'_i(y)$, where $B'_i(y) = \alpha_i \rightarrow_\top B_i(y)$, and $\alpha_i = A_i(x_0)$ is the compatibility of the rule premise with the input.

If the i-th rule is given as "If x_1 is A_{i1} and ... and x_k is A_{ik} then y is B_i", input x_0 is in fact a vector and we have $\alpha_i = \min(A_{i1}(x_{0_1}), ..., A_{ik}(x_{0_k}))$. This formalism has a very important restriction: if there exists an input x such that two rules R_1 and R_2 are fired with $\alpha_1 = \alpha_2 = 1$ then we must have $\exists y, B_1(y) = B_2(y) = 1$ to guarantee that the result is consistent, i.e. that $output(K, \{x_0\})$ is normalized for every acceptable x_0. Following [9], here inference is performed using Goguen residuated implication operator, defined as $a \rightarrow_{\top\Pi} b = 1$ if $a \leq b$ and $a \rightarrow_{\top\Pi} b = b/a$ otherwise.

For example, let us consider the subordination effect on the secretary for state $s_a = (.4, .7, 0)$. Fuzzy terms C and N are addressed for input variable $s_1 = .4$ and the same happens with $s_2 = .7$. Therefore, four rules are fired, addressing output terms $\{N, G, VG, EG\}$. Since we obtain $\alpha = .5$ for rules with output VG and EG and $\alpha = .125$ for those with N and G, an inconsistent result is produced.

To deal with an inconsistency, following [7], we make the fuzzy output terms less specific by the use of a similarity relation. From a result coming from [5], we only need to make consistent the terms that are the furthest apart from each other. Thus, in our example with $s_a = (.4, .7, 0)$, we only need to consider N and EG.

The application of a similarity relation S on a fuzzy term A, denoted by $S \circ A$, creates a "larger" term $approximately\_A$. Formally, we have

$$(S \circ A)(x) = \sup_{y \in U} \min(S(x, y), A(y)).$$

Here, we use the linear similarity relation family $S_\lambda(x, y) = \max(0, 1 - \lambda^{-1} \cdot |x - y|)$, where $\lambda > 0$.

In our example with $s_a = (.4, .7, 0)$, the smallest parameter that makes N and EG compatible with each other, given the values for s_1 and s_2, is $\lambda = 2.27$; consequently $effect_{sub}(Secretary, s_a) = < 5.7, 7.8, 8.2 >$ (see Figure 3).

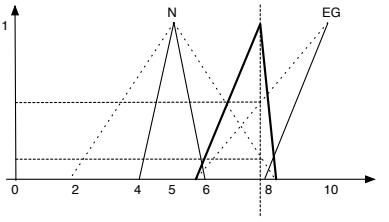


Fig. 3. Inference of fuzzy gradual rules and similarity relations

The result of the inference for $effect_{aut}(Secretary, s_a)$ is obtained straightforwardly as $N =< 4, 5, 6 >$. We finally compute the secretary's payoff for s_a as $payoff = 1 \otimes effect_{aut} \oplus 9 \otimes effect_{sub} =< 55.5, 75.7, 80.4 >$. The results for the states considered the Travel-Tours example are given in Table 5 below.

Table 5. Payoffs in the Travel-Tours example using rule bases

state	Director	Secretary
$s_a = (.4, .7, 0)$	<41.8, 49.3, 58.2>	<55.5, 75.7, 80.4>
$s_b = (.4, 0, .8)$	<71, 86.5, 88>	<47, 53.5, 59>
$s_c = (.4, .7, .8)$	<69.8, 84.3, 86.2>	<53.5, 74.7, 79.4>

With the rule bases and fuzzy terms used here we obtained the same behaviour as in the crisp case: state s_a is good for the secretary but bad for the director (considering his other payoffs), whereas s_b is very good for the director but not so good for the secretary (considering her other payoffs), and state s_c is a compromise.

Note that the results obtained using the extension principle are compatible with those obtained through the rule-based one: the preference order induced on states s_a, s_b and s_c, in what regards the secretary and the director, is the same in the two approaches even though the specific numerical values differ. Moreover, in both approaches there is no need to compare the numerical values issued for the secretary and director, as preconized by the foundations of welfare economics after the 30s, based on ordinal and interpersonally non-comparable utility information [13].

5 Conclusion

We presented here two approaches to extend a fragment modeling the Sociology of Organized Action to the fuzzy framework. The first approach is useful when it is possible to express assessments directly through mathematical expressions, and is modeled using the extension principle. A drawback of this approach is that functions involving non-linearities have to be approximated to make computation efficient. The other approach employs fuzzy rule bases and is useful when the assessments can only be given in qualitative means.

As future work, we intend to address the cases in which i) a resource can be controlled by more than one actor and ii) an actor can be responsible for the limits on a resource controlled by someone else. We also intend to address the modeling of the strategies of actors according to a bounded rationality, important in the implementation of social games.

References

1. Binmore, K.: Fun and Games: a Text on Game Theory. D.C. Health and Company (1992)
2. Crozier, M.: The Bureaucratic Phenomenon, University of Chicago Press (1964)
3. Crozier, M., Friedberg, E.: Organizations and Collective Action: our Contribution to Organizational Analysis. In: Bacharrach, S.B., Gagliardi, P., Mundel, B. (eds.) Studies of Organiszations in the European Tradition. Series "Research in the Sociology of Organizations" Greenwich Corn., vol. 13, pp. 71–93. Jay-Press (1995)
4. Dubois, D., Prade, H.: Possibility theory: an approach to computerized processing of uncertainty. Plenum Press (1988)
5. Dubois, D., Prade, H., Ughetto, L.: Checking the coherence and redundancy of fuzzy knowledge bases. IEEE Trans. on Fuzzy Systems 5(3), 398–417 (1997)
6. Encyclopedia Universalis (2004)
7. Godo, L., Sandri, S.: Dealing with covering problems in fuzzy rule systems by similarity-based extrapolation. In: Proc. FuzzIEEE'02, Honolulu (USA) (2002)
8. Mailliard, M., Audras, S., Marina, C.: MultiAgents Systems based on Classifiers for the Simulation of Concrete Action Systems. In: EUMAS: the 1st EUropean Workshop on Multi-Agent Systems, Oxford University (UK) (2003)
9. Sandri, S., Sibertin-Blanc, C., Torra, V.: A multicriteria fuzzy system using residuated implication operators and fuzzy arithmetic. In: MDAI'07: Modeling Decisions for Artificial Intelligence, Kitakyushu (Jp) (to be presented)
10. Sibertin-Blanc, C., Amblard, F., Mailliard, M.: A coordination framework based on the Sociology of Organized Action. In: Boissier, O., Padget, J., Dignum, V., Lindemann, G., Matson, E., Ossowski, S., Sichman, J.S., Vázquez-Salceda, J. (eds.) Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems. LNCS (LNAI), vol. 3913, pp. 3–17. Springer, Heidelberg (2006)
11. Simon, H.: The sciences of the artificial, 3rd edn. MIT Press, Cambridge (1996)
12. Smets, P.: L'agence Travel-Tours, `http://homepages.ulb.ac.be/~psmets1/travel.pdf`
13. Suzumura, K.: Introduction. In: Arrow, K., Sen, A.K., Suzumura, K. (eds.) Handbook of Social Choice and Welfare. 1. Series Handbooks in Economics 19, North-Holland

An Axiomatization of Conditional Possibilistic Preference Functionals

Didier Dubois[1], Hélène Fargier[1], and Barbara Vantaggi[2]

[1] Institut de Recherche en Informatique de Toulouse, France
{dubois,fargier}@irit.fr
[2] Dip. Me.Mo.Mat. Università "La Sapienza" di Roma, Italy
vantaggi@dmmm.uniroma1.it

Abstract. The aim of the paper is to extend the Savage like axiomatization of possibilistic preference functionals in qualitative decision theory to conditional acts, so as to make a step towards the dynamic decision setting. To this end, the de Finetti style approach to conditional possibility recently advocated by Coletti and Vantaggi is exploited, extending to conditional acts the basic axioms pertaining to conditional events.

1 Introduction

The natural counterparts to the expected utility criterion is the pair of possibilistic optimistic and pessimistic criteria, originally introduced by Yager [21] and Whalen [19] respectively. These criteria were axiomatized in the setting of Von-Neuman and Morgenstern theory, based on the comparison of possibilistic lotteries by Dubois et al. [7] and in the Savagean setting of acts under uncertainty by Dubois Prade and Sabbadin [10]. Later on, Giang and Shenoy [13] introduced a possibilistic criterion without pessimism nor optimism assumptions, using a bipolar qualitative scale concatenating the possibility and necessity scales. In this setting, each act is evaluated by a pair of qualitative values. The criterion is like the pessimistic one when all potential consequences are bad, and like the optimistic ones where all consequences are good. More recently Paul Weng [20] showed how to axiomatize this criterion in the Savage setting. All the above works propose a foundation to qualitative decision making in a static world. But the important issue of a qualitative decision theory when new input information can be received was left open.

In classical decision theory, this question turns out to be an easy one because of the sure thing principle. When the input information is obtained under the form of a true event A, the expected utility of acts comes down to restricting the acts to states of nature where this event is true, making the consequences outside A of all acts identical, regardless of what these common consequences are. Namely, if the preference relation indexed by the sure event is representable by an expected utility, then the same holds for the preference relation indexed by the event A, whenever the conditioning event A is not null. It comes down to changing the subjective probability into a conditional probability. This method

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 803–815, 2007.

can fail when the possible event A is null, that is, indifferent to the constant zero.

The issue of conditional qualitative criteria is more difficult, because in possibility theory the sure thing principle fails. As a consequence, the axiomatization of conditional possibilistic criteria must be reconsidered from scratch. It can be done either using a set of conditional preference relations on acts, or using a single preference relation on conditional acts. In the first approach, preference relations are indexed by an event that represents the information context in which the decision takes place. Additional axioms must be found in order to explain how preference relations indexed by different events can interplay. In the second approach one considers any act that takes place in a given information context. From an uncertainty-theoretic point of view it comes down to studying conditional set-functions not as a derived notion built from the unconditional ones, but as a primitive notion. This approach to uncertainty measures is the one adopted by de Finetti for probability theory, in order to allow for conditioning on hypothetical events with probability zero. This is the path followed in this paper. Recently Coletti and Vantaggi [2,3] introduced this approach in qualitative possibility theory, thus extending to the conditional setting the comparative possibility relation first proposed by Lewis [17], and retrieved by Dubois [6] as an ordinal account of Zadeh's possibility theory [22] in the spirit of comparative probability also originally proposed by de Finetti. The merit of qualitative conditional possibility after Coletti and Vantaggi is to provide an answer to conditioning on non-empty events of possibility zero, thus capturing a more general concept of conditioning (including some other proposals already studied in literature).

The aim of this paper is to bridge the gap between qualitative conditional possibility and the axiomatization of possibilistic preference functionals, thus paving the way toward possibilistic decision under uncertainty in a dynamic epistemic environment.

2 Decision-Theoretic Approach to Possibility Theory

A decision problem under uncertainty will be cast in the usual framework: we consider set S of states and a set $\mathcal{X}$ of potential consequences of decisions. States encode possible situations, states of affairs, etc. An act is viewed as a mapping f from the state space to the consequence set, namely, in each state $s \in S$, an act f produces a well-defined result $f(s) \in \mathcal{X}$. The decision maker must rank acts without knowing what is the current state of the world in a precise way. In qualitative decision theory, S is finite, and so is generally $\mathcal{X}$. n will denote the number of states in S. The consequences of an act can often be ranked in terms of their relative appeal: some consequences are judged better than others. This is often modeled by means of a numerical utility function u which assigns to each consequence $x \in \mathcal{X}$ a utility value $u(x) \in \boldsymbol{R}$.

The most widely found assumption is that there is a probability distribution p on S, and the most usual decision rule is based on the expected utility criterion.

When no information about the current state is available, the maximin criterion ranks acts according to its worst consequence:

$$W_u^-(f) = \min_{s \in S} u(f(s)). \tag{1}$$

Clearly this criterion has the major defect of being extremely pessimistic. Absolute qualitative approaches rely on extensions of Wald's criterion. The possibilistic qualitative criterion is based on a utility function u on $\mathcal{X}$ and a possibility distribution π on S [22], both mapping on the same totally ordered scale L. The ordinal value $\pi(s)$ represents the relative plausibility of state s. Here, L is equipped with its involutive order-reversing map $\mathbf{n}$; in particular $\mathbf{n}(1_L) = 0_L, \mathbf{n}(0_L) = 1_L$. So, $\mathbf{n}(\pi(s))$ represents the degree of potential surprise in case the state of the world is s [18]. In particular, $\mathbf{n}(\pi(s)) = 1_L$ for impossible states. A pessimistic criterion $W_{\pi,u}^-(f)$ is proposed [19,9] of the form :

$$W_{\pi,u}^-(f) = \min_{s \in S} \max(\mathbf{n}(\pi(s)), u(f(s))) \tag{2}$$

The value of $W_{\pi,u}^-(f)$ is small as soon as there exists a highly plausible state ($\mathbf{n}(\pi(s)) = 0_L$) with low utility value. This criterion is actually a prioritized extension of the Wald maximin criterion $W_u^-(f)$. The latter is recovered in case of total ignorance, ie. when $\pi(s) = 1_L$ for all $s \in S$. The decisions are again made according to the merits of acts in their worst consequences, now restricted to the most plausible states defined by a compromise between belief and utility expressed in the min-max expression.

The optimistic counterpart to this criterion [22,21] is:

$$W_{\pi,u}^+(f) = \max_{s \in S} \min(\pi(s), u(f(s))). \tag{3}$$

The optimistic and pessimistic possibilistic criteria are particular cases of a more general criterion based on the Sugeno integral (see [14]):

$$S_{\gamma,u}(f) = \max_{\lambda \in L} \min(\lambda, \gamma(F_\lambda)) \tag{4}$$

where $F_\lambda = \{s \in S, u(f(s)) \geq \lambda\}$, γ is a monotonic set function that reflects the decision-maker's attitude in front of uncertainty: $\gamma(A)$ is the degree of confidence in event A. The possibilistic criterion $W_{\pi,u}^+$ is obtained when γ is the possibility measure based on π ($\gamma(A) = \max_{s \in A} \pi(s)$), and $W_{\pi,u}^-$ is obtained when γ is the corresponding necessity measure ($\gamma(A) = \min_{s \notin A} \mathbf{n}(\pi(s))$ [1]).

We consider Sugeno integral and possibilistic criteria in the scope of Savage theory. Let us denote $\succeq$ a complete and transitive preference relation among acts of $\mathcal{X}^S$: $\succ$ will denote its strict part ($f \succ g \iff f \succeq g$ and $\neg(g \succeq f)$) and $\simeq$ will denote its symmetric part ($f \simeq g \iff f \succeq g$ and $g \succeq f$).

We denote fAh the act identical to f on a subset A and to h on its complementary: $\forall s, fAh(s) = f(s)$ if $s \in A, h(s)$ if $s \notin A$. The possibilistic criteria $W_{\pi,u}^+$ and $W_{\pi,u}^-$ satisfy a weak version of the sure-thing principle:

[1] Indeed, it is easy to show that $S_{\gamma,u}(f) = \max_{s \in S} \min(u(f(s)), \gamma(F_{u(f(s))}))$ is equal to $\min_{s \in S} \max(u(f(s)), \gamma(F_{\overline{u(f(s))}}))$, where $F_{\overline{\lambda}} = \{s \in S, u(f(s)) > \lambda\}$ [14].

Axiom WP2: $\forall A, \forall f, g, h, h', fAh \succ gAh \Rightarrow fAh' \succeq gAh'$.

Let us denote by $\succeq_P$ the utility ordering of consequences that derives from $\succeq$: $x \succeq_P y \iff f_x \succeq f_y$. where f_x (resp. f_y) is the constant act that concludes to consequence x (resp. y) for any state.

The rankings of acts obtained by a Sugeno integral satisfy the following weak version of Savage postulate P3:

Axiom WP3: $\forall A \subseteq S, \forall x, y \in \mathcal{X}, \forall f, x \succeq_P y$ implies $xAf \succeq yAf$.

But the converse may be false for events the plausibility of which is lower than the utility degree of x and y (the plausibility degree of A is in this case so negligible with respect to the utility of x and y that A is considered as null in this context).

The basic properties of Sugeno integrals exploit *disjunctive and conjunctive combinations* of acts. Let act $f \wedge g$ be the one always producing the worst consequences of f and g in each state, while $f \vee g$ always makes the best of them:

$$f \wedge g(s) = f(s) \text{ if } g(s) \succeq_P f(s) \text{ and } g(s) \text{ otherwise} \tag{5}$$

$$f \vee g(s) = f(s) \text{ if } f(s) \succeq_P g(s) \text{ and } g(s) \text{ otherwise} \tag{6}$$

They are union and intersection of fuzzy sets viewed as acts. Obviously, $S_{\gamma,u}(f \wedge g) \leq \min(S_{\gamma,u}(f), S_{\gamma,u}(g))$ and $S_{\gamma,u}(f \vee g) \geq \max(S_{\gamma,u}(f), S_{\gamma,u}(g))$ from weak Pareto monotonicity. These properties hold with equality whenever f or g is a constant act and are then characteristic of Sugeno integrals for monotonic aggregation operators [16]. Actually, these properties can be expressed by means of axioms, called restricted conjunctive and disjunctive dominance (RCD and RDD) on the preference structure $(\mathcal{X}^S, \succeq)$:

- Axiom RCD: if f is a constant act, $f \succ h$ and $g \succ h$ imply $f \wedge g \succ h$
- Axiom RDD: if f is a constant act, $h \succ f$ and $h \succ g$ imply $h \succ f \vee g$.

For instance, RCD means that limiting from above the potential utility values of an act g, that is better than another one h, to a constant value that is better than the utility of act h, still yields an act better than h. This is in contradiction with expected utility theory and strongly counterintuitive in the context of economic theory, with a continuous consequence set $\mathcal{X}$. However the range of validity of qualitative decision theory is precisely when both $\mathcal{X}$ and S are finite and steps in the finite value scale are far from each other.

This setting enables the axiomatization of Sugeno integrals in the style of Savage to be carried out. The following representation theorem holds:

Theorem 1 *[11]: A preference structure $(\mathcal{X}^S, \succeq)$ is a non-trivial weak order that satisfies WP3, RCD and RDD if and only if there exists a finite chain of preference levels L, an L-valued monotonic set-function γ, and an L-valued utility function u on $\mathcal{X}$, such that $f \succeq g$ if and only if $S_{\gamma,u}(f) \geq S_{\gamma,u}(g)$.*

The pessimistic criterion $W^-_{\pi,u}(f)$ can be axiomatized by strengthening axiom RCD into conjunctive dominance as follows [10]:

Axiom CD: $\forall f, g, h, f \succ h$ and $g \succ h$ imply $f \wedge g \succ h$.

Changing RDD into CD implies that the set-function γ is a necessity measure [10] and so, $S_{\gamma,u}(f) = W^{-}_{\pi,u}(f)$ for some possibility distribution π. Similarly, the criterion $W^{+}_{\pi,u}(f)$ can be axiomatized by strengthening axiom RDD into disjunctive dominance as follows:

Axiom DD: $\forall f, g, h, h \succ f$ and $h \succ g$ imply $h \succ f \vee g$.

Changing RCD into DD implies that the set-function γ is a possibility measure and so, $S_{\gamma,u}(f) = W^{+}_{\pi,u}(f)$ for some possibility distribution π. In order to figure out why axiom CD leads to a pessimistic criterion, let us notice here that CD can be equivalently replaced by the following property:

$$(\text{PESS})\forall A \subseteq S, \forall f, g, fAg \succ g \text{ implies } g \succeq gAf. \tag{7}$$

Similarly, the following optimistic counterpart to (7) can serve as a substitute to axiom DD for the representation of criterion $W^{+}_{\pi,u}$:

$$(\text{OPT})\forall A \subseteq S, \forall f, g, g \succ fAg \text{ implies } gAf \succeq g. \tag{8}$$

3 Qualitative Conditional Possibility

The notion of conditioning in possibility theory is a problem of long-standing interest. Starting from a triangular norm (t-norm) T various definitions of T-conditional possibility have been given [8]. In the following we use the axiomatic definition proposed in [1], restricted to the t-norm minimum:

Definition 1. *Let $S = \{s_1, ..., s_n\}$ be a state space and $\boldsymbol{E} = \boldsymbol{B} \times \boldsymbol{H}$ where $\boldsymbol{B}$ is a finite algebra of subsets of S, $\boldsymbol{H} \subseteq \boldsymbol{B} \setminus \{\emptyset\}$ an additive class of non-empty subsets of S (closed with respect to finite unions). A function $\Pi : \boldsymbol{E} \to [0, 1]$ is a qualitative conditional possibility if it satisfies the following properties:*

1. $\Pi(E|H) = \Pi(E \wedge H|H)$, for every $E \in \boldsymbol{B}$ and $H \in \boldsymbol{H}$;
2. $\Pi(\cdot|H)$ is a possibility measure, for every $H \in \boldsymbol{H}$;
3. $\forall H, E \wedge H \in \boldsymbol{H}$ and $E, F \in \boldsymbol{B}, \Pi(E \wedge F|H) = \min\left(\Pi(E|H), \Pi(F|E \wedge H)\right)$.

Condition *2* requires that, for every conditioning event $H \in \mathbf{H}$, the function $\Pi(\cdot|H)$ is a possibility, so it is normalized. A characterization of qualitative conditional possibilities in terms of a class of unconditional possibilities on the algebra $\mathbf{B}$ was given in [2]. An analogous result for T-conditional possibility, with T a strictly increasing t-norm, is in [12] and it is in the same line as the characterization theorem of conditional probabilities in de Finetti approach [5]. In both cases the conditional possibility $\Pi(\cdot|H)$ is not singled-out by the possibility of its conditioning event H, but its value is ruled by the values of other possibilities $\Pi(\cdot|E \wedge H)$, for suitable events E. It turns out that a conditional possibility cannot always be derived from just one "unconditional" possibility. The value $\Pi(E|H)$ follows directly from $\Pi(E \wedge H)$ and $\Pi(H)$ just in the case

$\Pi(E \wedge H) < \Pi(H)$. Note that in such a case Definition 1 coincides with the one given by Dubois and Prade in [8], which is based on the minimum specificity principle and consists in taking for $\Pi(E|H)$ the greatest solution to the equation $\Pi(E \wedge H) = \min\{x, \Pi(H)\}$, that is $\Pi(E \wedge H)$ when $\Pi(E \wedge H) < \Pi(H)$ and 1_L otherwise. Definition 1 is more general than the latter.

For example, consider the following conditional possibility on $\mathbf{E} = \mathbf{B} \times \{H, \Omega\}$, with $S = \{s_1, s_2, s_3\}$ and $H = s_1 \vee s_2$:

$$\Pi(\{s_1\}) = \Pi(\{s_2\}) = 0.2; \Pi(\{s_1\}|H) = 0.6. \tag{9}$$

Note that the rules of possibility theory imply $\Pi(s_1 \vee s_2) = 0.2$ and $\pi(s_3) = \Pi(F) = 1$, where $s_3 \in F \in \mathbf{B}$. Similarly, the conditional constraint implies $\pi(s_2|H) = \Pi(H|H) = 1$. Let $\Pi_0 = \Pi_{|\Omega}$. It is a solution to both constraints in (9) but the equation $\Pi_0(s_1) = \min(x, \Pi_0(s_1 \vee s_2))$ does not define a unique conditional possibility. The solution to system (9) is a pair of unconditional possibilities (i.e. $\{\Pi_0, \Pi_1\}$, where $\pi_1(s_1) = 0.6; \pi_1(s_2) = 1; \pi_1(s_3) = 0$). Moreover, Π_1 is the unique solution to $\Pi(\{s_1\}|H) = 0.6$ on referential H.

Characterizations of ordinal relations $\preceq$ on a set of conditional events $\mathbf{E} = \mathbf{B} \times \mathbf{H}$ representable by qualitative conditional possibilities Π (i.e. for any $A|H, B|K \in \mathbf{E}$, $A|H \preceq B|K \leftrightarrow \Pi(A|H) \leq \Pi(B|K)$) have been provided in [3,4]. In the sequel we recall the main results.

Definition 2. *A binary relation $\preceq$ on conditional events $A|H \in \boldsymbol{E}$ is called comparative conditional possibility iff the following conditions hold:*

1. *$\preceq$ is a weak order;*
2. *for any $H, K \in \boldsymbol{H}$, $\emptyset|H \sim \emptyset|K \prec H|H \sim K|K$;*
3. *for any $A, B \in \boldsymbol{B}$ and $H, B \wedge H \in \boldsymbol{H}$, $A \wedge B|H \preceq A|B \wedge H$ and moreover if either $A \wedge B|H \prec B|H$ or $B|H \sim H|H$, then $A \wedge B|H \sim A|B \wedge H$;*
4. *for any $H \in \boldsymbol{H}$ and any $A, B, C \in \boldsymbol{B}$, $A|H \preceq B|H \Rightarrow (A \vee C)|H \preceq (B \vee C)|H$.*

Condition (3) requires that in the context of the new information "B" the degree of belief in an event A cannot be less than the degree of belief in $A \wedge B$ before supposing that B occurs. Moreover, if the new information B is less surprising than $A \wedge B$ in the context H, or even totally unsurprising, the occurrence of B cannot change the degree of belief in A in the context H. Condition (4) is essentially the one proposed by Dubois [6], just rewritten conditioned on the hypothesis H. Moreover, condition (4) is equivalent (see [3]), under transitivity, to $A|H \preceq B|K$ and $C|H \preceq D|K \Rightarrow (A \vee C)|H \preceq (B \vee D)|K$.

Theorem 1. *[3]: For a binary relation $\preceq$ on $\boldsymbol{E} = \boldsymbol{B} \times \boldsymbol{H}$ the following statements are equivalent:*

i. $\preceq$ is a comparative conditional possibility;
ii. there exists a qualitative conditional possibility Π on $\boldsymbol{E}$ representing $\preceq$.

Obviously, among the comparative conditional possibilities there are also the ordinal relations representable by conditional possibilities satisfying the minimum

specificity principle, more precisely those satisfying a reinforcement of condition *3* of Definition 2, that is

(*sc*) for every $A, B \in \mathbf{B}$ and $H, B \wedge H \in \mathbf{H}$, $(A \wedge B)|H \preceq A|(B \wedge H)$ and moreover if $A \wedge B \wedge H \neq \emptyset$ and $(A \wedge B)|H \sim B|H$, then $A|(B \wedge H) \sim H|H$.

4 Qualitative Conditional Possibilistic Preference Functional: Optimistic Case

Let S be a finite set of states, $\mathbf{B}$ be the power set of S and $\mathbf{H} \subseteq \mathbf{B} \setminus \{\emptyset\}$ be an additive class containing S.[2]

Given a set of consequences $\mathcal{X}$, a conditional act $f|H$ is formed by a pair: an act f and an event $H \in \mathbf{H}$. The event H in $f|H$ is not just representing a given fact, but it is an uncertain hypothetical event whose truth value may be unknown. It expresses the idea of choosing decision f in case H were true, not actually doing it when H occurs. It differs from an unconditional act of the form fHg even if the value of $f|H$ and fHg is equal to $f(s) \in \mathcal{X}$ for any state $s \in H$. Indeed, for $s \notin H$, the value of $f|H$ is *undetermined* (following the terminology of de Finetti).

Let x^* and x_* be the best and the worst consequences (according to a given preference) in $\mathcal{X}$. Moreover the event $E \in \mathbf{B}$ is in bijection with the binary act taking the best value x^* when E is true and the worst value x_* when E is false.

A qualitative conditional decision model consists of a conditional possibility $\Pi : \mathbf{B} \times \mathbf{H} \to L$, a utility function u on the consequences in $\mathcal{X}$ with $u(x^*) = 1_L$ and $u(x_*) = 0_L$. A *conditional* possibilistic optimistic criterion takes the form:

$$v^*(f|H) = \max_{s \in H} \{\min\{u(f(s)), \Pi(s|H)\}\}.$$

Note that the above model is such that, for any $H \in \mathbf{H}$,

$$v^*(x^*|H) = \max_{s \in H} \min\{u(x^*), \Pi(s|H)\} = \max_{s \in H} \Pi(s|H) = \Pi(H|H) = 1 = u(x^*)$$

and $v^*(x_*|H) = \max_{s \in H} \min\{u(x_*), \Pi(s|H)\} = u(x_*) = 0 = \Pi(\emptyset|H)$.

4.1 Axioms for the Qualitative Conditional Model: Optimistic Case

Given a preference on the set of conditional acts $f|H$ with consequences on $\mathcal{X}$, we consider the following conditions:

1. $\preceq$ is a non-trivial weak order on $\mathcal{F} = \mathcal{X}^S \times \mathbf{H}$;
2. for any consequences $x, y \in \mathcal{X}$ such that $x >_p y$ and for any $H, K \in \mathbf{H}$ one has

$$y|H \sim y|K \text{ and } y|H \prec x|H;$$

[2] This assumption could be dropped.

3. (WP3) if x, y are consequences in $\mathcal{X}$ such that $x \geq_p y$, then $(yAh)|H \preceq (xAh)|H$ for any act h and any $A \in \mathbf{B}$ and $H \in \mathbf{H}$;
4. (OPT) for any $f|H, g|H$ and for any $A \in \mathbf{H}$
$$(fAg)|H \prec f|H \Rightarrow f|H \preceq (gAf)|H$$
5. (RCD) for any constant act f_x
$$f|H \prec g|H \text{ and } f|H \prec f_x|H \Rightarrow f|H \prec (g \wedge f_x)|H;$$
6. for any $x, y \in \mathcal{X}$ such that $x >_p y$ and for any $A, B \in \mathbf{B}$ and $H, B \wedge H \in \mathbf{H}$,
$$(x(A \wedge B)y)|H \preceq (xAy)|B \wedge H$$
and moreover if $(x(A \wedge B)y)|H \prec (xBy)|H$ or $(xBy)|H \sim x|H$, then
$$(xA \wedge By)|H \sim (xAy)|B \wedge H.$$

Conditions 1,3,4,5 are trivial generalizations of axioms proposed in qualitative possibilistic decision theory. Conditions 2 and 6 compare conditional acts with different conditioning events and are generalizations of those proposed in Definition 2 [3] for comparative conditional possibility. Note that the approach reduces to axioms of qualitative possibilistic decision theory when fixing the conditioning event. Condition 2 is useful to compare constant acts with different conditioning events, stating that the merit of a constant act is not affected by the conditioning event. Note that condition 6 is actually a reinforcement of the axiom proposed in [3], by requiring its validity for all the conditional binary acts, i.e. conditional acts of the form $(xAy)|H$ having, when H is true, two consequences $x >_p y \in \mathcal{X}$, more precisely x when A is true and y when A is false. Actually, condition 6 involves all the pairs of constant acts $x >_p y$. The first part of the condition suggests the decision-maker always prefers a more precise context $(B \wedge H)$ for the act involving event A. Indeed, conditional act $(x(A \wedge B)y)|H$ is risky since a bad consequence obtains when $A \wedge B^c$ occurs, while this possibility is ruled out by act $(xAy)|B \wedge H$, in the context $B \wedge H$. The second part of condition 6 can be explained as follows: if improving consequence y into x on $A \wedge B^c$ makes the act $(x(A \wedge B)y)|H$ more attractive, assuming B is true in the context H makes act $x(A \wedge B)y$ indifferent to xAy. Moreover, the same conclusion is reached if, in context H, event B is considered so likely that act xBy is like the constant act f_x.

It is easy to see that the conditional possibilistic optimistic criterion satisfies these properties:

Proposition 1. *A conditional optimistic criterion induces a preference relation satisfying conditions* 1 *to* 6.

Proof. Condition 1 holds since v^* is valued on a totally ordered scale. The validity of conditions $3, 4, 5$ follows from [10]. Conditions $2, 6$ follow from [3].

Since $\preceq$ is a total preorder on $\mathcal{F}$, its restriction to constant acts induces the same type of relation $\geq_p$ on $\mathcal{X}$. Thus, among the consequences we can find the best and the worst acts, denoted by x^* and x_*, respectively. Now we can reconstruct the conditional possibilistic optimistic criterion using the following steps:

Lemma 1. *Let $\preceq$ be a preference relation on $\mathcal{F}$ satisfying conditions* $1, 3, 4$. *Then,* $(fAg)|H \prec f|H \Rightarrow f|H \preceq (hAf)|H$ *for any* h.

Proof. If $(fAg)|H \prec f|H$, then $(fAx_*)|H \prec f|H$.
Suppose there exists an act h such that $(hAf)|H \prec f|H$. Then $(x_*Af)|H \preceq (hAf)|H \prec f|H$, so a contradiction for condition 4 (OPT) arises.

The two results in the sequel are trivial generalizations of the ones given for the unconditional case in [10]

Lemma 2. *Let $\preceq$ be a preference relation on $\mathcal{F}$ satisfying conditions* $1, 3, 4$. *If $h = f \vee g$, then $h|H \sim f|H$ or $h|H \sim g|H$.*

Lemma 3. *Let $\preceq$ be a preference relation on $\mathcal{F}$ satisfying conditions* $1, 3, 4, 5$. *If $h = f \wedge f_x$, where f_x is a constant act with value x, then $h|H \sim f|H$ or $h|H \sim f_x|H$.*

The next step retrieves a comparative conditional possibility on events:

Theorem 2. *Let $\preceq$ be a preference relation on $\mathcal{F}$ satisfying conditions* 1 *to* 6. *Then the restriction of $\preceq$ on the acts of the form $x^*Ex_*|H$ with $E \in \boldsymbol{B}$ and $H \in \boldsymbol{H}$ is a comparative conditional possibility.*

Proof. We consider the bijection introduced in [11] between acts of the form x^*Ex_* and events E, where x^* and x_* are the best and the worst consequences (according to a given preference) in $\mathcal{X}$. It follows from condition 2 that, for any $H, K \in \mathbf{H}$, $\emptyset|H \sim \emptyset|K \prec H|H \sim K|K$. Condition 1 implies that the restriction of $\preceq$ is a non-trivial weak order on the set of conditional events $E|H \in \mathbf{B} \times \mathbf{H}$.

From condition 6 it follows that $A \wedge B|H \preceq A|B \wedge H$, for any $A, B \in \mathbf{B}$ and $H, B \wedge H \in \mathbf{H}$. Moreover, when $A \wedge B|H \prec B|H$ or $B|H \sim H|H$, it follows $A \wedge B|H \sim A|B \wedge H$.

Taking condition 4 into play, and letting $f = x^*Bx_*$, $g = x^*Ax_*$ and $h = x^*A^cx_*$ one has $fAg = x^*(A \wedge B)x_*$, and $hAf = x^*(A^c \wedge B)x_*$, then condition 4 implies that if $(A \wedge B)|H \prec B|H$, then by Lemma 1 $A^c \wedge B|H \sim B|H$. Hence, $A \wedge B|H \sim B|H$ or $A^c \wedge B|H \sim B|H$, which is equivalent under monotonicity to $A|H \preceq B|H \Rightarrow A \vee C|H \preceq B \vee C|H$ (see [6]).

Corollary 1. *Let $\preceq$ be a preference relation on $\mathcal{F}$ satisfying conditions* 1 *to* 6. *Then the restriction of $\preceq$ on the acts of the form $x^*Ex_*|H$ with $E \in \boldsymbol{B}$ and $H \in \boldsymbol{H}$ is representable by a qualitative conditional possibility.*

Proof. From Theorem 2 it follows that the restriction of a preference relation on the conditional events, which satisfies condition 1 to 6, is a comparative conditional possibility, then the main result in [3] implies that it is representable by a qualitative conditional possibility.

Theorem 3. *Let S be a finite set of states, $\boldsymbol{B}$ be the power set of S and $\boldsymbol{H} \subseteq \boldsymbol{B} \setminus \{\emptyset\}$ an additive class of events such that $S \in \boldsymbol{H}$. Let $\preceq$ be a preference over $\mathcal{F} = \mathcal{X}^S \times \boldsymbol{H}$, which satisfies conditions* 1 *to* 6. *Then, there exists a finite totally*

ordered scale L, a utility function $u : \mathcal{X} \to L$, a qualitative conditional possibility $\Pi : \boldsymbol{B} \times \boldsymbol{H} \to L$, and a function $V : \mathcal{F} \to L$, which represents $\preceq$. Moreover V is of the form

$$V(f|H) = \max_{s \in S} \{\min\{u(f(s)), \Pi(s|H)\}\}.$$

Proof. 1. *Building a utility scale* Since $\mathcal{F}$ is finite, from condition 1 it follows that there exists a function V that represents $\preceq$, taking values in a finite linear ordered scale L with smallest and the greatest values 0_L and 1_L, respectively. The value associated to the conditional act $f|H$ (and to its equivalent acts) is $V(f|H)$. Since $S \in \mathbf{H}$, take a constant act f_x and let $u(x) = V(f_x)$. Moreover, due to point-wise preference $u(x_*) = 0_L$ and $u(x^*) = 1_L$. By condition 2, since $f_x|H \sim f_x|K$, for any $H, K \in \mathbf{H}$ (and so $f_x \sim f_x|S$), it follows that, for any $H \in \mathbf{H}, V(f_x|H) = u(x)$ [3].

2. *Building a qualitative conditional possibility* The construction of a qualitative conditional possibility $\Pi(\cdot|\cdot)$ on $(x^* A x_*)|H$ follows from Theorem 2 and Corollary 1.

3. *Computation of the utilities of acts of the form $xEy|H$* Consider a conditional act of the form $(xEx_*)|H = (x^* E x_*) \wedge f_x|H$, from Lemma 3 one has $(xEx_*)|H \sim (x^* E x_*)|H$ or $(xEx_*)|H \sim f_x|H$, then, (see point 1 of the proof)

$$V(xEx_*|H) = V(x^* E x_*|H) = \Pi(E|H) \text{ or } V(xEx_*|H) = V(f_x|H) = u(x).$$

Since $(xEx_*)|H \preceq (x^* E x_*)|H$ and $(xEx_*)|H \preceq f_x|H$, then

$$V(xEx_*|H) = \min\{u(x), \Pi(E|H)\}.$$

A conditional binary act $(xEy)|H$, with $x \geq_p y$ (without loss of generality), can be written as $((xEx_*) \vee f_y|H$ and by Lemma 2 it follows $(xEy)|H \sim (xEx_*)|H$ or $(xEy)|H \sim f_y|H$, moreover $(xEx_*)|H \preceq (xEy)|H$ and $\mathbf{y}|H \preceq (xEy)|H$, hence

$$V(xEy|H) = \max\{V(xEx_*|H), V(f_y|H)\} = \max\{V(xEx_*|H), u(y)\}.$$

More generally, by decomposing any act through its value on states s, we get $f|H = \bigvee_{s \in H} f(s)\{s\}x_*|H$, then it follows

$$V(f|H) = \max_{s \in H}\{V(f(s)\{s\}x_*|H)\} = \max_{s \in S} \min\{\Pi(s|H), u(f(s))\}\}.$$

Note that we can also write $V(f|H) = \max_{s \in H} \min\{\Pi(s|H), u(f(s))\}\}$ since $\Pi(s|H) = 0$ if $s \notin H$.

[3] If $S \notin \mathbf{H}$, consider $H^o = \bigvee_{H \in \mathbf{H}} H$ and put, for any constant act $f_x, u(x) = V(f_x|H^o)$. Thus, as in the previous case, $u(x_*) = 0_L$ and $u(x^*) = 1_L$ and again from condition 2 it follows $V(f_x|H) = u(x)$.

4.2 Conditional Possibilistic Preference Functional: The Pessimistic Case

A pessimistic qualitative possibilistic criterion presupposes a conditional necessity function $N : \mathbf{B} \times \mathbf{H} \to L$, a utility u on the consequences in $\mathcal{X}$ with $u(x^*) = 1_L$ and $u(x_*) = 0_L$. A *conditional* pessimistic criterion is of the form:

$$v_*(f|H) = \min_{s \in H} \max(u(f(s)), N(\{s\}^c|H)).$$

Note that the above functional is such that, for any $H \in \mathbf{H}$,

$$v_*(x^*|H) = \min_{s \in H} \max\{u(x^*), N(\{s\}^c|H)\} = u(x^*) = 1 = N(H|H)$$

and $v_*(x_*|H) = \min_{s \in H} \max\{u(x_*), N(\{s\}^c|H)\} = \min_{s \in H} N(\{s\}^c|H) = 0$.

Directly axiomatizing the pessimistic qualitative possibilistic criterion would require a drastic modification of condition 6 since the latter extends properties of conditional possibility orderings, not necessity orderings. It cannot be done here for lack of space. However, the necessity function can be expressed as $N(\{s\}^c|H) = \mathbf{n}(\pi(s|H))$ where $\mathbf{n}(\cdot)$ is the order reversing map in L. Then the pessimistic criterion can be expressed in terms of an expression close to the one of the optimistic criterion, since

$$\mathbf{n}(v_*(f|H)) = \max_{s \in H} \min(\mathbf{n}(u(f(s))), \pi(s|H))$$

which lays bare its meaning: $v_*(f|H)$ is all the higher as there is no plausible state with high disutility $\mathbf{n}(u(f(s))$. So it maybe axiomatized by directly from pessimism axioms, preserving condition 6, and constructing a max-min disutility preferential $D(f \mid H) = \mathbf{n}(v_*(f|H))$ from a disutility function $\delta = \mathbf{n}(u)$ on $\mathcal{X}$ such that $\delta(x^*) = 0_L$ and $\delta(x_*) = 1_L$. The connection between a generalization of the specific condition introduced in [4] for comparative conditional necessities and the above model needs to be analyzed.

5 Conclusion

This paper takes a first step toward extending the scope of qualitative decision theory to conditional events, thus making it possible to update qualitative optimistic and pessimistic preference functionals. The rescaling function for the representation of uncertainty is rather simple since only the most plausible states allowed by the context induced by the new information are mapped to the top value of the scale. However conditioning on a null event may end up with a different possibiity distribution. Our results are only a first step, and several improvements could be envisaged

– Condition 6 is expressed in terms of binary acts, and is formally a copy of conditional possibility ordering axioms. It would be much more convincing to derive this condition from an axiom involving general acts.

– As the meaning of conditional acts $f|H$ may look difficult to grasp, it may sound more natural to axiomatize the conditional criteria in the setting of a preference relation indexed by the context, like $\prec_H$ (H not empty). For instance, $f \prec_H g$ may be another way of denoting $f|H \prec g|H$ [15]. But clearly, encoding the statement $f|A \prec g|B$ using relations of the form $\prec_H$ is not obvious, and the language of conditional acts is likely to be richer.

Among more advanced lines of further research that can be considered, the extension of the framework to more general set-functions, the application of these criteria to qualitative Markov decision processes and the study of dynamic consistency can be envisaged.

References

1. Bouchon-Meunier, B., Coletti, G., Marsala, C.: Independence and Possibilistic Conditioning. Ann. of Math. and Artif. Intell. 35, 107–124 (2002)
2. Coletti, G., Vantaggi, B.: Possibility Theory: Conditional Independence. Fuzzy Sets and Systems 157, 1491–1513 (2006)
3. Coletti, G., Vantaggi, B.: Comparative Conditional Possibilities. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 872–883. Springer, Heidelberg (2005)
4. Coletti, G., Vantaggi, B.: Comparative Models Ruled by Possibility and Necessity: A Conditional World. Int. Journal of Approximate Reasoning 45, 341–363 (2007)
5. de Finetti, B.: Sul Significato Soggettivo della Probabilità. Fundamenta Matematicae 17, 293–329 (1931)
6. Dubois, D.: Belief structures, Possibility Theory and Decomposable Confidence Measures on Finite Sets. Computers and AI (Bratislava) 5(5), 403–416 (1986)
7. Dubois, D., Godo, L., Prade, H., Zapico, A.: On the Possibilistic Decision Model: From Decision under Uncertainty to Case-based Decision. Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems 7, 631–670 (1999)
8. Dubois, D., Prade, H.: Possibility Theory. Plenum Press, New York (1988)
9. Dubois, D., Prade, H.: Possibility Theory as a Basis for Qualitative Decision Theory. In: Proc. IJCAI'95, Montréal, Canada, pp. 1924–1930 (1995)
10. Dubois, D., Prade, H., Sabbadin, R.: Decision-theoretic Foundations of Possibility Theory. European Journal of Operational Research 128, 459–478 (2001)
11. Dubois, D., Prade, H., Sabbadin, R.: Qualitative Decision Theory with Sugeno Integrals. In: Proc. of 14th Conf. on Uncertainty in AI (UAI'98), Madison, pp. 121–128. Morgan Kaufmann, San Francisco (1998)
12. Ferracuti, L., Vantaggi, B.: Independence and Conditional Possibilities for Strictly Monotone Triangular Norms. Internat. J. Intelligent Systems 21, 299–323 (2006)
13. Giang, P.H., Shenoy, P.P.: Two Axiomatic Approaches to Decision Making using Possibility Theory. European Journal of Operational Research 162, 450–467 (2005)
14. Grabisch, M., Murofushi, T., Sugeno, M. (eds.): Fuzzy Measures and Integrals. Physica-Verlag, Heidelberg (2000)
15. Lehmann, D.: Generalized Qualitative Probability: Savage revisited. In: Proc. 12th Conf. on Uncertainty in Artificial Intelligence1996, Portland, pp. 381–388. Morgan Kaufmann, San Francisco (1996)
16. Marichal, J.L.: On Sugeno Integral as an Aggregation Function. Fuzzy Sets and Systems 114, 347–365 (2000)

17. Lewis, D.: Counterfactuals. Basil Blackwell, London (1973)
18. Shackle, G.L.S.: Decision, Order and Time in Human Affairs. Cambridge University Press, Cambridge (1961)
19. Whalen, T.: Decision Making under Uncertainty with Various Assumptions about Available Information. IEEE Trans. on Systems, Man and Cybernetics 14, 888–900 (1984)
20. Weng, P.: An Axiomatic Approach in Qualitative Decision Theory with Binary Possibilistic Utility. In: ECAI, pp. 467–471 (2006)
21. Yager, R.R.: Possibilistic Decision Making. IEEE Trans. on Systems, Man and Cybernetics 9, 388–392 (1979)
22. Zadeh, L.A.: Fuzzy Sets as a Basis for a Theory of Possibility. Fuzzy Sets and Systems 1, 3–28 (1978)

Conflict Analysis and Merging Operators Selection in Possibility Theory

Weiru Liu

School of Electronics, Electrical Engineering and Computer Science,
Queen's University Belfast,
Belfast, BT7 1NN, UK
w.liu@qub.ac.uk

Abstract. In possibility theory, *the degree of inconsistency* is commonly used to measure the level of conflict in information from multiple sources after merging, especially conjunctive merging. However, as shown in [HL05,Liu06b], this measure alone is not enough when pairs of uncertain information have the same degree of inconsistency, since it is not possible to tell which pair contains information that is actually *better*, in the sense that the two pieces of information in one pair agree with each other more than the information does in other pairs. In this paper, we investigate what additional measures can be used to judge the *closeness* between two pieces of uncertain information. We deploy the concept of *distance between betting commitments* developed in DS theory in [Liu06a], since possibility theory can be viewed as a special case of DS theory. We present properties that reveal the interconnections and differences between the degree of inconsistency and the distance between betting commitments. We also discuss how to use these two measures together to guide the possible selection of various merging operators in possibility theory.

1 Introduction

Pieces of uncertain information that come from different sources often do not agree with each other completely. There can be many reasons for this, such as, inaccuracy in sensor data reading, natural errors occurring in experiments, and unreliability of sources. When inconsistent information needs to be merged, assessing the degree of conflict among information plays a crucial role when deciding which combination mode would suit the best [DP94].

In possibility theory, the two basic combination modes are *conjunctive* and *disjunctive*, each of which has some specific merging operators. Some conjunctive operators also have reinforcement effects and they are more suitable to combine information that is highly consistent. In general, conjunctive operators are advised to combine information that is reliable and consistent and disjunctive operators are advised to merge inconsistent information [BDKP02]. The degree of inconsistency of merged information is widely used to judge how consistent that two (or multiple) pieces of possibilistic information are. Clearly, this value is not sufficient when multiple pairs of uncertain information have the same degree of inconsistency. We need additional approaches to measuring the degree of agreement (or conflict) between two pieces of possibilistic

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 816–827, 2007.

information in order to accurately decide which merging operator is more suitable, especially when an reinforcement operator is to be used.

In this paper, we take the advantage that possibility theory can be regarded as a special case of Dempster-Shafer theory and investigate how the degree of agreement (or conflict) between possibilistic uncertain information can be assessed by the concept of *distance between betting commitments* proposed in [Liu06a]. We particularly study the relationships between these two measures and are able to provide the following findings.

First, when a pair of possibilistic uncertain information is totally contradictory with each other, both measures give the same result, i.e., the maximum value of conflict. Second, when a pair of possibilistic uncertain information appears to be consistent, i.e., the degree of inconsistency is zero, the range of values of the distance between betting commitments can vary from zero to almost one. This finding is important since it tells that these two measures reveal two different aspects of the information involved. Third, when the degree of inconsistency is sufficiently large, the distance between betting commitments increases proportionally, that is the latter is a function of the former. Based on these findings, we are able to provide a set of more detailed guidelines as which conjunctive (or reinforcement) merging operator is more suitable for combining a given pair of uncertain information.

We will proceed as follows: in Section 2, we review the basics in possibility theory and DS theory and their connections. In Section 3, we investigate the relationships between the degree of inconsistency and the distance between betting commitments. In Section 4, we first review the general guidelines about how to select a merging operator in possibility theory (or possibilistic logic), we then provide a set of refined guidelines for this purpose. Finally in Section 5, we summarize the main contributions of the paper.

2 Preliminaries

2.1 Possibility Theory

Possibility theory (or possibilistic logic) is a popular choice for representing uncertain information (or knowledge) ([DP82,BDP97], etc). At the semantic level, a basic function in possibility theory is a **possibility distribution** denoted as π which assigns each possible world in set Ω a value in $[0, 1]$ (or a set of graded values).

From a possibility distribution, a possibility measure (denoted as Π) and a necessity measure (denoted as N) can be derived as

$$\Pi(A) = \mathsf{max}(\{\pi(\omega)|\omega \in A\}) \text{ and } N(A) = 1 - \Pi(\bar{A}), \bar{A} = \Omega \setminus A \qquad (1)$$

The former estimates to what extent the true event is believed to be in the subset and the latter evaluates the degree of necessity that the subset is true.

For a given π, if there exists $\omega_0 \in \Omega$ such that $\pi(\omega_0) = 1$, then π is said to be normal, otherwise, π is not normal. The value $1 - \mathsf{max}_{\omega \in \Omega}\pi(\omega)$ is called **the degree of inconsistency** of the information (or possibility distribution).

In possibility theory, the two families of merging operators are conjunctive and disjunctive. Examples of conjunctive operators are min, product and linear product

and an example of disjunctive operator is max. Given two possibility distributions π_1 and π_2, the semantic results of applying these operators are $\forall\omega \in \Omega$, $\pi_{\mathsf{min}}(\omega) = \mathsf{min}(\pi_1(\omega), \pi_2(\omega))$, $\pi_{\times}(\omega) = \pi_1(\omega) \times \pi_2(\omega)$, $\pi_{\otimes}(\omega) = \mathsf{max}(0, \pi_1(\omega) + \pi_2(\omega) - 1)$, and $\pi_{\mathsf{max}}(\omega) = \mathsf{max}(\pi_1(\omega), \pi_2(\omega))$, where we use $\times$ and $\otimes$ for product and linear product operators respectively.

2.2 Basics of Dempster-Shafer Theory

In the Dempster-Shafer theory of evidence (DS theory) [Sha76], a piece of uncertain information is represented by a **basic probability assignment** (or called a **mass function**) m on a set (Ω) containing mutually exclusive and exhaustive solutions to a question. Ω is called the **frame of discernment**.

A mass function $m : 2^{\Omega} \rightarrow [0, 1]$ satisfies $m(\emptyset) = 0$ and $\sum_{A\subseteq\Omega} m(A) = 1$ (though condition $m(\emptyset) = 0$ is not strictly required in the Transferable Belief Model (TBM) [SK94]).

From m, a **belief function**, $Bel(A) : 2^{\Omega} \rightarrow [0, 1]$ is defined as $Bel(A)=\Sigma_{B\subseteq A} m(B)$. When $m(A) > 0$, A is referred to as a *focal element* of the belief function (by abuse of language, we simply say A is a focal element of mass function m in the rest of the paper). A **plausibility function** $Pl : 2^{\Omega} \rightarrow [0, 1]$ from m is defined as $Pl(A) = \Sigma_{B\cap A\neq\emptyset} m(B)$.

Two mass functions from distinct sources are usually combined using Dempster's combination rule. The rule is stated as follows.

Definition 1. *Let m_1 and m_2 be mass functions, and let $m_1 \oplus m_2$ be the combined mass function.*

$$m_1 \oplus m_2(C) = \frac{\Sigma_{A\cap B=C}\ (m_1(A) \times m_2(B))}{1 - \Sigma_{A\cap B=\emptyset}\ (m_1(A) \times m_2(B))}$$

when $\Sigma_{A\cap B=\emptyset}\ (m_1(A) \times m_2(B)) \neq 1$.

$\sum_{B\cap C=\emptyset} m_1(B)m_2(C)$ is the mass of the combined belief assigned to the emptyset before normalization and we denote it as $m_{\oplus}(\emptyset)$. In the following, whenever we use $m_{\oplus}(\emptyset)$, we always associate it with this explanation unless otherwise explicitly stated.

Definition 2. *[Sme04] Let m be a mass function on Ω. Its associated* **pignistic probability function** $\mathsf{BetP}_m : \Omega \rightarrow [0, 1]$ *is defined as*

$$\mathsf{BetP}_m(\omega) = \sum_{A\subseteq\Omega,\omega\in A} \frac{m(A)}{|A|}$$

where $|A|$ is the cardinality of subset A.

The transformation from m to BetP_m is called the **pignistic transformation**. In the original definition [Sme04], when $m(\emptyset) \neq 0$, $m(A)$ is replaced by $\frac{m(A)}{1-m(\emptyset)}$ in the above definition. Furthermore, $\mathsf{BetP}_m(A) = \sum_{\omega\in A} \mathsf{BetP}_m(\omega)$ for $A \subseteq \Omega$.

Definition 3. *([Liu06a]) Let m_1 and m_2 be two mass functions on frame Ω and let BetP_{m_1} and BetP_{m_2} be their corresponding pignistic probability functions respectively. Then*

$$\mathsf{difBetP}^{m_2}_{m_1} = \mathsf{max}_{A\subseteq\Omega}(|\mathsf{BetP}_{m_1}(A) - \mathsf{BetP}_{m_2}(A)|)$$

is called the **distance between betting commitments** *of the two mass functions.*

Value $(|\mathsf{BetP}_{m_1}(A) - \mathsf{BetP}_{m_2}(A)|)$ is *the difference between betting commitments to A* from the two sources. The distance of betting commitments is therefore the maximum extent of the differences between betting commitments to all the subsets. $\mathsf{difBetP}^{m_2}_{m_1}$ is simplified as $\mathsf{difBetP}$ when there is no confusion as which two mass functions are being compared.

2.3 DS Theory Versus Possibility Theory

It has long been recognized that possibility theory is a special case of DS theory in the sense that from a possibility distribution, a mass function with nested focal elements can be recovered from it (e.g., [DP82]). In this case, a belief function is a necessity measure and a plausibility function is a possibility measure. The actual procedure to recover a mass function (and hence a belief function) is stated in the following definition.

Definition 4. *([DP82,DP88]) Let π be a possibility distribution on frame of discernment Ω and be normal. Let the set of values $\pi(\omega_i)$ be $\{\alpha_i | i = 1, ..., p\}$ and they are arranged as $\alpha_1 = 1 \geq \alpha_2 \geq \alpha_3, ..., \geq \alpha_p > 0$ and $\alpha_{p+1} = 0$. Let*

1. *$A_i = \{\omega | \pi(\omega) \geq \alpha_i\}$ for $i = 1, 2, ..., p$, then subsets $A_1, A_2, .., A_p$ are nested;*
2. *$m(A_i) = \pi(\omega_i) - \pi(\omega_{i+1})$ for $i = 1, 2, ..., p$, where $\omega_i \in A_i, \omega_{i+1} \in A_{i+1}$.*

Then m is a mass function recovered from π with focal elements A_i ($i = 1..., p$).

Example 1. Let π be a possibility distribution on $\Omega = \{\omega_1, ..., \omega_4\}$ where

$$\pi(\omega_1) = 0.7, \pi_2(\omega_2) = 1.0, \pi_2(\omega_3) = 0.8, \pi_2(\omega_4) = 0.7$$

Then the focal elements are $A_1 = \{\omega_2\}$, $A_2 = \{\omega_2, \omega_3\}$, and $A_3 = \Omega$. The corresponding mass function is $m(A_1) = 0.2$, $m(A_2) = 0.1$, and $m(A_3) = 0.7$.

3 Relationship Between $\mathsf{Inc}(\pi)$, $\mathsf{difBetP}$ and $m_\oplus(\emptyset)$

Since $\mathsf{Inc}(\pi)$, $\mathsf{difBetP}$ and $m_\oplus(\emptyset)$ are developed for measuring inconsistency/conflict in possibility theory and DS theory respectively, and these two theories have some interconnections, we study formally the relationships among these three values.

Proposition 1. *([Liu06b]) Let π be a possibility distribution on frame of discernment Ω and be normal. Let BetP_m be the pignistic probability function of the corresponding mass function m derived from π. Then $\mathsf{BetP}_m(\omega_i) \geq \mathsf{BetP}_m(\omega_j)$ iff $\pi(\omega_i) \geq \pi(\omega_j)$.*

This proposition says that the more plausible a possible world is, the more betting commitment it carries. It is consistent with the *ordinal faithfulness* [Dub06] where a probability distribution preserves the ordering of possibilities of elementary events[1].

[1] It should be noted that in [Dub06], *ordinal faithfulness* refers to the preservation of the ordering of elementary events after transforming a probability distribution to a possibility distribution. Since obtaining BetP_m from a π satisfies this feature, we think it is worth to mention it here.

Example 2. (Con't Example 1) *Following Example 1, the pignistic probability function for the given possibility distribution is*

$\mathsf{BetP}_m(\omega_1) = 0.7/4$; $\mathsf{BetP}_m(\omega_2) = 0.2 + 0.1/2 + 0.7/4$;
$\mathsf{BetP}_m(\omega_3) = 0.1/2 + 0.7/4$; $\mathsf{BetP}_m(\omega_4) = 0.7/4$.

That is $\mathsf{BetP}_m(\omega_2) > \mathsf{BetP}_m(\omega_3) > \mathsf{BetP}_m(\omega_1) = \mathsf{BetP}_m(\omega_4)$ *which is consistent with the ordering of* $\pi(\omega_2) > \pi(\omega_3) > \pi(\omega_1) = \pi(\omega_4)$.

Proposition 2. *Let* π_1 *and* π_2 *be two possibility distributions on frame of discernment* Ω *and be normal. Let* π_{min}, $\pi_\times$ *and* $\pi_\otimes$ *be their merged results using the* min*, the* product*, and the* linear product *operators respectively. Then the following properties hold*

$$\mathsf{Inc}(\pi_{\mathsf{min}}) = 1 \textit{ iff } \mathsf{Inc}(\pi_\times) = 1 \textit{ iff } \mathsf{Inc}(\pi_\otimes) = 1$$

$$\mathsf{Inc}(\pi_{\mathsf{min}}) = 0 \textit{ iff } \mathsf{Inc}(\pi_\times) = 0 \textit{ iff } \mathsf{Inc}(\pi_\otimes) = 0$$

The proof of this proposition is straightforward and it enables us to prove the following propositions by using the min as the representative of conjunctive operators.

Proposition 3. *([Liu06b]) Let* π_1 *and* π_2 *be two possibility distributions on* Ω *and be normal. Let* $\pi_\wedge$ *be their conjunctively merged possibility distribution. Assume* m_1 *and* m_2 *are the mass functions derived from* π_1 *and* π_2 *respectively. Then the following properties hold*

1. $\mathsf{Inc}(\pi_\wedge) = 0$ *iff* $m_\oplus(\emptyset) = 0$
2. $\mathsf{Inc}(\pi_\wedge) = 1$ *iff* $m_\oplus(\emptyset) = 1$
3. $\mathsf{Inc}(\pi_\wedge) > 0$ *iff* $m_\oplus(\emptyset) > 0$

If we have two pairs of possibility distributions and we use $\pi^1_\wedge$ and $\pi^2_\wedge$ to denote their conjunctively merged possibility distributions, then $\mathsf{Inc}(\pi^1_\wedge) \geq \mathsf{Inc}(\pi^2_\wedge)$ does not imply $m^1_\oplus(\emptyset) \geq m^2_\oplus(\emptyset)$ in general, where $m^1_\oplus$ and $m^2_\oplus$ are the combined mass functions from the two pairs of mass functions derived from corresponding possibility distributions. This is demonstrated by Example 4 (in Section 4) where the two sets of possibility distributions have the same degree of inconsistency (0.2) but with different values assigned to the emptyset after combination (0.07 versus 0.23).

Proposition 4. *Let* π_1 *and* π_2 *be two possibility distributions and normal and let their conjunctively combined possibility distribution be* $\pi_\wedge$*. Furthermore, let* m_1 *and* m_2 *be their corresponding mass functions. Then we have the following property*

$$\mathsf{Inc}(\pi_\wedge) = 1 \textit{ iff } \mathsf{difBetP}^{\mathsf{m}_2}_{\mathsf{m}_1} = 1$$

Proof. We take $\pi_\wedge = \pi_{\mathsf{min}}$ below without losing generality (see Proposition 2).

We first prove that $\mathsf{Inc}(\pi) = 1$ implies $\mathsf{difBetP}^{\mathsf{m}_2}_{\mathsf{m}_1} = 1$.

Let π_1 and π_2 be two possibility distributions, where π_{min} is the conjunctively merged distribution using min. When $\mathsf{Inc}(\pi_{\mathsf{min}}) = 1$, π_1 and π_2 are totally inconsistent, then for any $\omega \in \Omega$ either $\pi_1(\omega) = 0$ or $\pi_2(\omega) = 0$ or both. Let A_p and A_q be the largest focal

elements of m_1 and m_2 respectively, then $A_p \cap A_q = \emptyset$, and both $\mathsf{BetP}_{m_1}(A_p) = 1$ and $\mathsf{BetP}_{m_2}(A_q) = 1$ hold. So we have $\mathsf{BetP}_{m_1}(A_p) - \mathsf{BetP}_{m_2}(A_p) = 1$, since we must have $\mathsf{BetP}_{m_2}(A_p) = 0$ when $\mathsf{BetP}_{m_2}(A_q) = 1$ (remember BetP is a probability function). Therefore $\mathsf{difBetP}^{\mathsf{m_2}}_{\mathsf{m_1}} = 1$ must be true.

Next, we prove that $\mathsf{difBetP}^{\mathsf{m_2}}_{\mathsf{m_1}} = 1$ implying $\mathsf{Inc}(\pi) = 1$. When $\mathsf{difBetP}^{\mathsf{m_2}}_{\mathsf{m_1}} = 1$, there exists a subset $A \subset \Omega$ such that $\mathsf{BetP}_{m_2}(A) = 1$ and $\mathsf{BetP}_{m_1}(A) = 0$ (or vise versa). This means that A is the largest focal element for m_2 which implies $\forall \omega \in A$, $\pi_2(\omega) \neq 0$ and $\forall \omega \notin A, \pi_2(\omega) = 0$. On the other hand, $\mathsf{BetP}_{m_1}(A) = 0$ tells us that $\forall \omega \in A$, $\pi_1(\omega) = 0$. Therefore, we have

$\forall \omega \in A, \pi_{\mathsf{min}}(\omega) = \mathsf{min}(\pi_1(\omega), \pi_2(\omega)) = 0,$ since $\pi_1(\omega) = 0$, and
$\forall \omega \notin A, \pi_{\mathsf{min}}(\omega) = \mathsf{min}(\pi_1(\omega), \pi_2(\omega)) = 0,$ since $\pi_2(\omega) = 0$.
That is $\forall \omega \in \Omega, \pi_{\mathsf{min}}(\omega) = 0$. So $\mathsf{Inc}(\pi_{\mathsf{min}}) = 1$, and so is $\mathsf{Inc}(\pi_\wedge) = 1$. ◇

Propositions 3 and 4 together tell us that if two pieces of information contradict with each other completely, any of the three measures (i.e., $\mathsf{Inc}(\pi)$, $m_\oplus(\emptyset)$, or $\mathsf{difBetP}^{m_2}_{m_1}$) is sufficient to quantitatively justify it.

In general, $\mathsf{Inc}(\pi_\wedge) = 0 \Rightarrow \mathsf{difBetP}^{\mathsf{m_2}}_{\mathsf{m_1}} = 0$ does not hold as shown below.

Example 3. Let two possibility distributions be

$$\pi_1(\omega_1) = 1.0, \pi_1(\omega_2) = 0.1, \pi_1(\omega_3) = 1.0, \pi_1(\omega_4) = 0.8;$$

$$\pi_2(\omega_1) = 1.0, \pi_2(\omega_2) = 0.9, \pi_2(\omega_3) = 0.2, \pi_2(\omega_4) = 0.1.$$

Then the degree of inconsistency between them is 0 *if they are merged conjunctively. Their corresponding mass functions are*

$$m_1(\{\omega_1, \omega_3\}) = 0.2, m_1(\{\omega_1, \omega_3, \omega_4\}) = 0.7, m_1(\{\Omega\}) = 0.1;$$

$$m_2(\{\omega_1\}) = 0.1, m_2(\{\omega_1, \omega_2\}) = 0.7, m_2(\{\omega_1, \omega_2, \omega_3\}) = 0.1, m_2(\{\Omega\}) = 0.1.$$

As we can see at least $\mathsf{BetP}_{m_2}(\omega_2) - \mathsf{BetP}_{m_1}(\omega_2) > 0$*, so* $\mathsf{difBetP}^{\mathsf{m_2}}_{\mathsf{m_1}} = 0$ *does not hold.*

Proposition 5. *Let* π_1 *and* π_2 *be two possibility distributions on* Ω *and normal, and let their conjunctively combined possibility distribution be* $\pi_\wedge$*. Furthermore, let* m_1 *and* m_2 *be their corresponding mass functions. When* $\mathsf{Inc}(\pi_\wedge) = 0$ *we have*

$$0 \leq \mathsf{difBetP}^{\mathsf{m_2}}_{\mathsf{m_1}} \leq \frac{(n-1)}{n}, \textit{where } n = |\Omega| .$$

Proof. When two possibility distributions π_1 and π_2 are identical, $\mathsf{Inc}(\pi_\wedge) = 0$ must be true. Also, they generate the same mass function, and the same pignistic probability function, so $\mathsf{difBetP}^{\mathsf{m_2}}_{\mathsf{m_1}} = 0$ holds in this situation. We have shown that $\mathsf{difBetP}^{\mathsf{m_2}}_{\mathsf{m_1}} > 0$ is possible when $\mathsf{Inc}(\pi_\wedge) = 0$ in Example 3, therefore, $0 \leq \mathsf{difBetP}^{\mathsf{m_2}}_{\mathsf{m_1}}$ is true for any pair of possibility distributions when $\mathsf{Inc}(\pi_\wedge) = 0$.

Now, we prove that $\mathsf{difBetP}^{\mathsf{m_2}}_{\mathsf{m_1}} \leq \frac{(n-1)}{n}$.

$\mathsf{Inc}(\pi_\wedge) = 0$ implies that there is at least one $\omega \in \Omega$ such that $\pi_1(\omega) = 1$ and $\pi_2(\omega) = 1$.

First, we consider a situation where there is only one element in Ω, denoted as ω_1 such that $\pi_1(\omega_1) = 1$ and $\pi_2(\omega_1) = 1$. We further assume that for all $\omega_i \in \Omega$, $\pi_1(\omega_i) = 0$ if $\omega_i \neq \omega_1$ and $\pi_2(\omega_i) = 1$. Then the two mass functions from these two possibility distributions are $m_1(\{\omega_1\}) = 1$ and $m_2(\Omega) = 1$. Therefore

$$\mathsf{difBetP}^{\mathsf{m}_2}_{\mathsf{m}_1} = \frac{(n-1)}{n}$$

because $\mathsf{BetP}_{m_1}(\Omega \setminus \{\omega_1\}) = 0$ and $\mathsf{BetP}_{m_2}(\Omega \setminus \{\omega_1\}) = \frac{(n-1)}{n}$.

Before proving that for any two possibility distributions that $\mathsf{difBetP}^{\mathsf{m}_2}_{\mathsf{m}_1} \leq \frac{(n-1)}{n}$ holds, we need to prove that for a positive integer $n > 2$, the following inequality is true

$$\frac{n-1}{n} > \frac{n-2}{n-1}$$

This is obvious since $(n-1)^2 > n(n-2)$. Therefore, we have

$$\frac{n-1}{n} > \frac{n-2}{n-1} > \frac{n-3}{n-2} > ... > \frac{1}{2} \tag{2}$$

Next, we proof that for any π_1 and π_2 with their $\mathsf{Inc}(\pi_\wedge) = 0$, $\mathsf{difBetP}^{\mathsf{m}_2}_{\mathsf{m}_1} \leq \frac{(n-1)}{n}$.

For this case, we still assume that $\pi_1(\omega_1) = 1$ and $\pi_2(\omega_1) = 1$, because we have the assumption $\mathsf{Inc}(\pi) = 0$ which means $\exists w \in \Omega$, such that $\pi_1(\omega) = \pi_2(\omega) = 1$. Without losing generality, we assume that $\mathsf{BetP}_{m_2}(w_1) \leq \mathsf{BetP}_{m_1}(w_1)$ (since m_1 and m_2 are symmetric) and we can also assume that there exists a subset A such that $\mathsf{difBetP}^{\mathsf{m}_2}_{\mathsf{m}_1} = \mathsf{BetP}_{m_2}(A) - \mathsf{BetP}_{m_1}(A)$ holds (otherwise if $\mathsf{difBetP}^{\mathsf{m}_2}_{\mathsf{m}_1} = \mathsf{BetP}_{m_1}(A') - \mathsf{BetP}_{m_2}(A')$, we let $A = \Omega \setminus A'$ and the equation still holds). Let the sets of focal elements for m_2 be A_1, .., A_p where $A_1 \subset A_2 \subset ... \subset A_p$ and let $A'_p = A_p \setminus \{\omega_1\}$, we get

$$\begin{aligned}
\mathsf{BetP}_{m_2}(A'_p) &= \frac{|A_1|-1}{|A_1|} m_2(A_1) + ... + \frac{|A_p|-1}{A_p} m_2(A_p) \\
&\leq \frac{|A_p|-1}{A_p} m_2(A_1) + ... + \frac{|A_p|-1}{A_p} m_2(A_p) \text{(see Equation 2)} \\
&= \frac{|A_p|-1}{A_p}(m_2(A_1) + ... + m_2(A_p)) \\
&\leq \frac{|A_p|-1}{A_p}, \text{ (since } m_2(A_1) + ... + m_2(A_p) \leq 1) \\
&\leq \frac{n-1}{n} \text{ (because } A_p \subseteq \Omega, \text{ where } |\Omega| = n)
\end{aligned} \tag{3}$$

Then $\mathsf{BetP}_{m_2}(A'_p)$ is the largest value possible among all $\mathsf{BetP}_{m_2}(B)$ where $B \subseteq \Omega \setminus \{\omega_1\}$.

When $w_1 \in A$, we have

$$\begin{aligned}
\mathsf{difBetP}^{\mathsf{m}_2}_{\mathsf{m}_1} &= \mathsf{BetP}_{m_2}(A) - \mathsf{BetP}_{m_1}(A) \\
&= \mathsf{BetP}_{m_2}(A \setminus \{w_1\}) + \mathsf{BetP}_{m_2}(\{w_1\})
\end{aligned}$$

$$\begin{aligned} & -\mathsf{BetP}_{m_1}(A \setminus \{w_1\}) - \mathsf{BetP}_{m_1}(\{w_1\}) \\ \leq\ & \mathsf{BetP}_{m_2}(A \setminus \{w_1\}) - \mathsf{BetP}_{m_1}(A \setminus \{w_1\}) \\ \leq\ & \mathsf{BetP}_{m_2}(A_p^{'}) - 0 \leq \frac{n-1}{n} \end{aligned} \tag{4}$$

When $w_1 \notin A$, $\mathsf{difBetP}^{\mathsf{m_2}}_{\mathsf{m_1}} = \mathsf{BetP}_{m_2}(A) - \mathsf{BetP}_{m_1}(A) \leq \mathsf{BetP}_{m_2}(A_p^{'}) - 0 \leq \frac{n-1}{n}$. That is, $\mathsf{difBetP}^{\mathsf{m_2}}_{\mathsf{m_1}} \leq \frac{n-1}{n}$ is true for any two possibility distributions. ◇

This proposition is important, since it tells us that two apparently totally consistent possibility distributions can be very different when we measure their distances between betting commitments to subsets. This means that the two distributions can have very different degrees of possibility assigned to some elements, though they totally agree on some other elements. Therefore, using $\mathsf{Inc}(\pi_\wedge)$ alone may not be accurate enough when assessing how consistent (close) that two possibility distributions are.

Proposition 6. *Let π_1 and π_2 be two possibility distributions and normal, and let their conjunctively combined possibility distribution be $\pi_\wedge$. Furthermore, let m_1 and m_2 be their corresponding mass functions. When $\mathsf{Inc}(\pi_\wedge) = \epsilon$ where ϵ is sufficiently large (like 0.8), we have*

$$\mathsf{difBetP}^{\mathsf{m_2}}_{\mathsf{m_1}} \geq 2\epsilon - 1$$

Proof. First we assume that the values of $\pi_1(\omega)$ for all $\omega \in \Omega$ are arranged as (see Definition 4)

$$1 \geq \alpha_1 \geq ... \geq \alpha_i ... \geq \alpha_n > 0$$

Let α_i be the smallest value in the above sequence such that $\alpha_i > 1 - \epsilon$, based on Definition 4, we have a focal element A_i for m_1 as

$$A_i = \{w | \pi_1(w) \geq \alpha_i\}$$

If the other focal elements obtained before A_i are A_1, ..., A_{i-1}, then according to Definition 4, we have

$$\Sigma_{j=1}^{i} m_1(A_j) = (1-\alpha_1) + (\alpha_1 - \alpha_2) + ... + (\alpha_{i-1} - \alpha_i) = 1 - \alpha_i > 1 - (1-\epsilon) = \epsilon$$

By Definition 2, we have $\mathsf{BetP}_{m_1}(A_i) \geq \Sigma_{j=1}^{i} m_1(A_j) > \epsilon$, since $A_i \subset A_{i+1}, ..., A_i \subset A_n$ where $A_{i+1}, ..., A_n$ are the remaining focal elements for m_1.

Similarly, for π_2, there is a subset B_j such that $\mathsf{BetP}_{m_2}(B_j) > \epsilon$. Because $\mathsf{Inc}(\pi_\wedge) = \epsilon$, $A_i \cap B_j = \emptyset$ must hold (otherwise, there is a $\omega \in A_i \cap B_j$ where $\mathsf{min}(\pi_1(\omega), \pi_2(\omega)) > 1 - \epsilon$, so $\mathsf{Inc}(\pi_\wedge) < \epsilon$ which contradict the original assumption).

Given that BetP_{m_2} is a probability function, $\mathsf{BetP}_{m_2}(A_i) \leq 1 - \epsilon$ must hold (because $\mathsf{BetP}_{m_2}(B_j) > \epsilon$ and $A_i \cap B_j = \emptyset$). Therefore, we have

$$\mathsf{BetP}_{m_1}(A_i) - \mathsf{BetP}_{m_2}(A_i) \geq \epsilon - (1-\epsilon) = 2\epsilon - 1$$

Since $\mathsf{difBetP}^{\mathsf{m_2}}_{\mathsf{m_1}} \geq \mathsf{BetP}_{m_1}(A_i) - \mathsf{BetP}_{m_2}(A_i)$ (see Definition 3), we have eventually $\mathsf{difBetP}^{\mathsf{m_2}}_{\mathsf{m_1}} \geq 2\epsilon - 1$. ◇

This proposition is meaningful when $\epsilon \geq 0.5$ and it states that the distance between betting commitments increases along with the increase of the degree of inconsistency.

4 Merging Operators Selection Criteria

4.1 Merging Operators in Possibility Theory

The fundamental classes of merging operators in possibility theory (or possibilistic logic) are **conjunctive** and **disjunctive** operators. Typical conjunctive operators are *minimum* ($\mathsf{min}(\pi_1(\omega), \pi_2(\omega))$), *product* ($\pi_1(\omega) \times \pi_2(\omega)$), and *linear product* ($\mathsf{max}(\pi_1(\omega) + \pi_2(\omega) - 1, 0)$), and their dual are the *maximum*, the *probabilistic sum* ($\pi_1(\omega) + \pi_2(\omega) - \pi_1(\omega)\pi_2(\omega)$), and the *bounded sum* ($\mathsf{min}(1, \pi_1(\omega)\pi_2(\omega))$). All these conjunctive and disjunctive operators are associative, so merging n possibility distributions can be done recursively, provided that there are no normalizations for the intermediate merging results.

Since some of these operators have special characteristics, two specialized classes of merging operators are further defined in [BDKP02], they are respectively **idempotent** and **reinforcement** operators. For example, the product and the linear product operators are also reinforcement operators, and minimum and maximum are idempotent operators. Furthermore, some **adaptive** operators were proposed which aim at integrating both conjunctive and disjunctive operators when neither of them alone is suitable for merging.

As discussed in [BDKP02], these five classes of operators are suitable for different situations. The conjunctive operators are used when it is believed that all the sources are reliable and these sources agree with each other. When there is a strong agreement among the sources, reinforcement operators are more suitable. On the other hand, the disjunctive operators are applied when it is believed that some sources are reliable but it is not known which of these sources are and when there is a high degree of conflict among sources. Idempotent operators can deal with redundant information where repeated information is only counted once. Since disjunctive operators are too cautious for merging sources with a low level of inconsistency, adaptive operators are suggested to integrate the behaviour of both conjunctive and disjunctive operators.

Among the three named conjunctive operators, it is well recognized ([DP01]) that the product operator is equivalent to the Dempster's combination rule for the computation of the plausibility of singletons. Therefore, the condition of applying Dempster's rule shall apply to this operator as well, i.e., the information comes from distinct or independent sources.

Although the above analysis provides a general guideline as which operator is suitable for what situation, there are no quantitative measures judging precisely when a particular operator should be selected. For example, what value of inconsistency is regarded as *a lower degree* of inconsistency?

We are interested in whether it is possible to provide some quantitative approaches to serving this purpose based on properties we have shown in the previous section, and hence we propose the following guidelines to recommend how to select a merging operator.

4.2 Merging Operators Selection Criteria

In the following, we use $\times$ and $\otimes$ to denote the *product* and the *linear product* operators.

Definition 5. *Let π_1 and π_2 be two possibility distributions and m_1 and m_2 be their corresponding mass functions. When* $\mathsf{Inc}(\pi_\wedge) = 0$,

- *if* $\mathsf{difBetP}^{\mathsf{m}_1}_{\mathsf{m}_2} = 0$*, then operator* $\otimes$ *is recommended if the information is from independent (distinct) sources; otherwise, operator* min *is recommended,*
- *if* $0 < \mathsf{difBetP}^{\mathsf{m}_1}_{\mathsf{m}_2} < \epsilon_1$*, then operator* $\times$ *is recommended if the information is from independent (distinct) sources; otherwise, operator* min *is recommended,*
- *if* $\epsilon_1 \leq \mathsf{difBetP}^{\mathsf{m}_1}_{\mathsf{m}_2} < \epsilon_2$*, then operator* $\times$ *can be applied with caution if the information is from independent (distinct) sources; otherwise, operator* min *is recommended,*
- *if* $\epsilon_2 \leq \mathsf{difBetP}^{\mathsf{m}_1}_{\mathsf{m}_2}$*, then operator* min *is recommended.*

where ϵ_1 *is sufficiently small (e.g.,* 0.3*) and* ϵ_2 *is sufficiently large (e.g.,* 0.8*).*

This definition shows that when $\mathsf{Inc}(\pi_\wedge) = 0$, we do not have to arbitrarily choose a conjunctive operator, the $\mathsf{difBetP}^{\mathsf{m}_1}_{\mathsf{m}_2}$ value provides additional information as whether a high reinforcement operator is more suitable. For example, when $\mathsf{difBetP}^{\mathsf{m}_1}_{\mathsf{m}_2} = 0$, it is more advisable to use $\otimes$ than $\times$ because the information is highly consistent. As stated in [DP94], the condition of choosing such a reinforcement operator is the independence of sources of information. When this condition cannot be guaranteed, min would be a safer option to use.

Definition 6. *Let* π_1 *and* π_2 *be two possibility distributions and* m_1 *and* m_2 *be their corresponding mass functions. When* $0 < \mathsf{Inc}(\pi_\wedge) < \epsilon$*,*

- *if* $\mathsf{difBetP}^{\mathsf{m}_1}_{\mathsf{m}_2} < \epsilon_1$*, then operator* $\times$ *is recommended if the information is from independent (distinct) sources; otherwise, operator* min *is recommended,*
- *if* $\epsilon_1 \leq \mathsf{difBetP}^{\mathsf{m}_1}_{\mathsf{m}_2} < \epsilon_2$*, then operator* $\times$ *can be applied with caution if the information is from independent (distinct) sources; otherwise, operator* min *is recommended,*
- *if* $\epsilon_2 \leq \mathsf{difBetP}^{\mathsf{m}_1}_{\mathsf{m}_2}$*, then operator* min *is recommended.*

where ϵ *is sufficiently small (e.g.,* 0.2*), and* ϵ_1 *and* ϵ_2 *are as defined in Definition 5.*

Example 4. let two pairs of possibility distributions be as given below.

$$\pi^1_1(\omega_1) = 0.7, \pi^1_1(\omega_2) = 0.8, \pi^1_1(\omega_3) = 1.0, \pi^1_1(\omega_4) = 0.6;$$

$$\pi^1_2(\omega_1) = 1.0, \pi^1_2(\omega_2) = 0.9, \pi^1_2(\omega_3) = 0.7, \pi^1_2(\omega_4) = 0.6.$$

and

$$\pi^2_1(\omega_1) = 0.1, \pi^2_1(\omega_2) = 0.8, \pi^2_1(\omega_3) = 1.0, \pi^2_1(\omega_4) = 0.1;$$

$$\pi^2_2(\omega_1) = 1.0, \pi^2_2(\omega_2) = 0.9, \pi^2_2(\omega_3) = 0.2, \pi^2_2(\omega_4) = 0.1.$$

We use $\pi^1_\wedge$ *and* $\pi^2_\wedge$ *to denote the combined possibility distributions from the two pairs using* min*. Their degrees of inconsistency are the same,* $\mathsf{Inc}(\pi^1_\wedge) = \mathsf{Inc}(\pi^2_\wedge) = 0.2$ *and this value suggests the application of a conjunctive operator based on Definition 6.*

The corresponding mass functions from the two pairs of possibility distributions are

$$m^1_1(\{\omega_3\}) = 0.2, m^1_1(\{\omega_2, \omega_3\}) = 0.1, m^1_1(\{\omega_1, \omega_2, \omega_3\}) = 0.1, m^1_1(\{\Omega\}) = 0.6;$$

$$m^1_2(\{\omega_1\}) = 0.1, m^1_2(\{\omega_1, \omega_2\}) = 0.2, m^1_2(\{\omega_1, \omega_2, \omega_3\}) = 0.1, m^1_2(\{\Omega\}) = 0.6;$$

and

$$m_1^2(\{\omega_3\}) = 0.2, m_1^2(\{\omega_2, \omega_3\}) = 0.7, m_1^2(\{\Omega\}) = 0.1;$$

$$m_2^2(\{\omega_1\}) = 0.1, m_2^2(\{\omega_1, \omega_2\}) = 0.7, m_2^2(\{\omega_1, \omega_2, \omega_3\}) = 0.1, m_2^2(\{\Omega\}) = 0.1.$$

For the first pair of mass functions, we have $\mathsf{difBetP}_{\mathsf{m}_1^1}^{\mathsf{m}_2^1} = 0.25$*, while for the 2nd pair we get* $\mathsf{difBetP}_{\mathsf{m}_1^2}^{\mathsf{m}_2^2} = 0.525$*. These two pairs show an obvious difference in* difBetP *values. The possibility distributions in the first pair are more consistent with each other than the two in the 2nd pair. However, this information is not reflected by* $\mathsf{Inc}(\pi_\wedge)$*.*

According to Definition 6, the first pair can be combined with the product operator ($\times$*) if the sources of information are distinct while it is better to merge the second pair with the minimum operator.*

Definition 7. *Let* π_1 *and* π_2 *be two possibility distributions and* m_1 *and* m_2 *be their corresponding mass functions. When* $\mathsf{Inc}(\pi_\wedge) \geq \epsilon$ *then a disjunctive operator is recommended to merge* π_1 *and* π_2*, where* ϵ *is sufficiently large, e.g.,* 0.8*.*

Example 5. Let two possibility distributions be

$$\pi_1(\omega_1) = 0.1, \pi_1(\omega_2) = 0.2, \pi_1(\omega_3) = 1.0, \pi_1(\omega_4) = 0.1;$$

$$\pi_2(\omega_1) = 1.0, \pi_2(\omega_2) = 0.2, \pi_2(\omega_3) = 0.2, \pi_2(\omega_4) = 0.1.$$

Let $\pi_\wedge$ *be the possibility distribution combining* π_1 *and* π_2 *with* min*, then the degree of inconsistency is* $\mathsf{Inc}(\pi_\wedge) = 0.8$ *which suggests a high degree of inconsistency. Therefore, the conjunctive operators are unlikely to be used.*

On the other hand, the two corresponding mass functions from the possibility distributions are

$$m_1(\{\omega_3\}) = 0.8, m_1(\{\omega_2, \omega_3\}) = 0.1, m_1(\{\Omega\}) = 0.1;$$

$$m_2(\{\omega_1\}) = 0.8, m_2(\{\omega_1, \omega_2, \omega_3\}) = 0.1, m_2(\{\Omega\}) = 0.1;$$

and $\mathsf{difBetP}_{\mathsf{m}_1}^{\mathsf{m}_2} = 0.72$ *which also hints a strong conflict among the two pieces of information.*

The *grey area* that the above three definitions did not cover is when $\epsilon_1 \leq \mathsf{Inc}(\pi_\wedge) \leq \epsilon_2$, such that $\epsilon_1 = 0.3, \epsilon_2 = 0.8$. In our future work, we will further investigate what other measures are needed in order to select a suitable merging operator for this situation.

5 Conclusion

In this paper, we have shown that additional approaches to measuring conflict among pieces of uncertain information are needed since the only measure used in possibility theory, e.g., the degree of inconsistency, is not sufficient.

We have studied how the distance between betting commitments developed in DS theory can be used to measure the inconsistency among pieces of uncertain information

in possibility theory. We have also established a set of properties to show the relationship between the degree of inconsistency and the distance between betting commitments between a pair of uncertain information. We conclude that these two measures tell us different aspects of the information and both values should be used to select a suitable merging operator. This investigation can be taken as the refinement of general discussions on merging operators selection in [BDKP02].

As pointed out in the paper, there is a *grey area* where it is not clear which merging operator is best suited. One of our future work is to explore other additional measures to see if some quantitative measures can be proposed to deal with these cases.

Acknowledgement. This work is supported by the EPSRC project with reference EP/D070864/1.

References

BDP97. Benferhat, S., Dubois, D., Prade, H.: From semantic to syntactic approach to information combination in possibilitic logic. In: Bouchon-Meunier, B. (ed.) Aggregation and Fusion of Imperfect Information, pp. 141–151. Physica Verlag, Heidelberg (1997)

BDKP02. Benferhat, S., Dubois, D., Kaci, S., Prade, H.: Possibilistic Merging and Distance-Based Fusion of Propositional Information. Ann. Math. Artificial Intelligence 34(1-3), 217–252 (2002)

DP82. Dubois, D., Prade, H.: On several representations of an uncertain body of evidence. In: Gupta, Sanchez. (eds.) Fuzzy Information and Decision Processes, pp. 167–181. North-Holland Publishing Company (1982)

DP88. Dubois, D., Prade, H.: Representation and combination of uncertainty with belief functions and possibility measures. Computational Intelligence 4, 244–264 (1988)

DP94. Dubois, D., Prade, H.: Possibility theory and data fusion in poorly informed environments. Control Engineering Practice 2(5), 811–823 (1994)

DP01. Dubois, D., Prade, H.: Possibility theory in information fusion. In: Riccia, Lenz, Kruse. (eds.) Data Fusion and Perception. CISM Courses and Lectures, vol. 431, pp. 53–76. Springer, Heidelberg (2001)

Dub06. Dubois, D.: Possibility theory and statistical reasoning. Computational Statistics and Data Analysis 51(1), 47–69 (2006)

HL05. Hunter, A., Liu, W.: Assessing the quality of merged information in possibilistic logic. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 415–425. Springer, Heidelberg (2005)

Liu06a. Liu, W.: Analyzing the degree of conflict among belief functions. Artificial Intelligence 170(11), 909–924 (2006)

Liu06b. Liu, W.: Measuring conflict between possibilistic uncertain information through belief function theory. In: Lang, J., Lin, F., Wang, J. (eds.) KSEM 2006. LNCS (LNAI), vol. 4092, pp. 265–277. Springer, Heidelberg (2006)

Sha76. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)

SK94. Smets, Ph., Kennes, K.: The transferable belief model. Artificial Intelligence 66(2), 191–234 (1994)

Sme04. Smets, Ph.: Decision making in the TBM: the necessity of the pignistic transformation. International Journal of Approximate Reasoning 38, 133–147 (2004)

Extending Description Logics with Uncertainty Reasoning in Possibilistic Logic

Guilin Qi[1], Jeff Z. Pan[2], and Qiu Ji[1]

[1] Institute AIFB, University of Karlsruhe, Germany
{gqi,qiji}@aifb.uni-karlsruhe.de
[2] Department of Computing Science, The University of Aberdeen
Aberdeen AB24 3FX
jpan@csd.abdn.ac.uk

Abstract. Possibilistic logic provides a convenient tool for dealing with inconsistency and handling uncertainty. In this paper, we propose possibilistic description logics as an extension of description logics. We give semantics and syntax of possibilistic description logics. We then define two inference services in possibilistic description logics. Since possibilistic inference suffers from the *drowning problem*, we consider a drowning-free variant of possibilistic inference, called linear order inference. Finally, we implement the algorithms for inference services in possibilistic description logics using KAON2 reasoner.

1 Introduction

Dealing with uncertainty in the Semantic Web has been recognized as an important problem in the recent decades. Two important classes of languages for representing uncertainty are probabilistic logic and possibilistic logic. Arguably, another important class of language for representing uncertainty is fuzzy set theory or fuzzy logic. Many approaches have been proposed to extend description logics with probabilistic reasoning, such as approaches reported in [12,10]. The work on fuzzy extension of ontology languages has also received a lot of attention (e.g., [18,17]). By contrast, there is relatively few work on combining possibilistic logic and description logic.

Possibilistic logic [5] or possibility theory offers a convenient tool for handling uncertain or prioritized formulas and coping with inconsistency. It is very powerful to represent partial or incomplete knowledge [4]. There are two different kinds of possibility theory: one is qualitative and the other is quantitative. Qualitative possibility theory is closely related to default theories and belief revision [7,3] while quantitative possibility can be related to probability theory and can be viewed as a special case of belief function [8].

The application of possibilistic logic to deal with uncertainty in the Semantic Web is first studied in [13] and is then discussed in [6]. When we obtain an ontology using ontology learning techniques, the axioms of the ontology are often attached with confidence degrees and the learned ontology may be inconsistent

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 828–839, 2007.

Table 1. Semantics of $\mathcal{ALC}$-concepts

Constructor	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
concept name	CN	$\mathsf{CN}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
general negation ($\mathcal{C}$)	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction ($\mathcal{U}$)	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
exists restriction ($\mathcal{E}$)	$\exists R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
value restriction	$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$

[11]. In this case, possibilistic logic provides a flexible framework to interpret the confidence values and to reason with the inconsistent ontology under uncertainty.

However, there exist problems which need further discussion. First, there is no formal definition of the semantics of possibilistic description logics. The semantic extension of possibilistic description logic is not trivial because we need negation of axioms to define the *necessity measure* from a *possibility distribution*. However, negation of axioms are not allowed in description logics. Second, there is no implementation of possibilistic inference in description logics.

In this paper, we present a possibilistic extension of description logics. We first give the syntax and semantics of possibilistic logics. We then define two inference services in possibilistic description logics. Since possibilistic inference suffers from the *drowning problem*, we consider a drowning-free variant of possibilistic inference, called linear order inference. Finally, we implement the algorithms for inference services in possibilistic description logics using KAON2 reasoner.

The rest of this paper proceeds as follows. Preliminaries on possibilistic logic and description logics are given in Section 2. Both syntax and semantics of possibilistic description logics are provided in Section 3. The inference services in possibilistic description logics are also given. After that, we provide algorithms for implementing reasoning problems in Section 4. Finally, we report preliminary results on implementation in Section 5.

2 Preliminaries

2.1 Description Logics

Due to the limitation of space, we do not provide a detailed introduction of Description Logics (DLs), but rather point the reader to [1]. A DL knowledge base $\Sigma = (\mathcal{T}, \mathcal{A})$ consists a set $\mathcal{T}$ (TBox) of concepts axioms[1] and a set $\mathcal{A}$ (ABox) of individual axioms. Concept axioms have the form $C \sqsubseteq D$ where C and D are (possibly complex) concept descriptions. The ABox contains *concept assertions* of the form $a : C$ where C is a concept and a is an individual name, and *role*

[1] TBox could contain some role axioms, for some expressive DLs such as $\mathcal{SHOIQ}$ [14].

assertions of the form $\langle a, b\rangle : R$, where R is a role, and a and b are individual names. A *concept description* (or simply *concept*) of the smallest propositionally closed DL $\mathcal{ALC}$ is defined by the following syntactic rules, where CN is a concept name, R is a role, C, C_1 and C_2 are concept descriptions:

$$\top \mid \bot \mid \mathsf{CN} \mid \neg C_1 \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \exists R.C \mid \forall R.C.$$

An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of the domain of the interpretation $\Delta^{\mathcal{I}}$ (a non-empty set) and the interpretation function $\cdot^{\mathcal{I}}$, which maps each concept name CN to a set $\mathsf{CN}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, each role name RN to a binary relation $\mathsf{RN}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ and each individual a to an object in the domain $a^{\mathcal{I}}$. The interpretation function can be extended to give semantics to concept descriptions (see Table 1). An interpretation $\mathcal{I}$ *satisfies* a concept axiom $C \sqsubseteq D$ (a concept assertion $a : C$ and a role assertion $\langle a, b\rangle : R$, resp.) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ ($a^{\mathcal{I}} \in C^{\mathcal{I}}$ and $\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in R^{\mathcal{I}}$ resp.). An interpretation $\mathcal{I}$ *satisfies* a knowledge base Σ if it satisfies all axioms in Σ; in this case, we say $\mathcal{I}$ is an interpretation of Σ. A knowledge base is *consistent* if it has an interpretation. A concept is unsatisfiable in Σ iff it is interpreted as an empty set by all the interpretation of Σ.

Most DLs are fragments of classical first-order predicate logic (FOL). An $\mathcal{ALC}$ knowledge bases can be translated to a $\mathcal{L}^2$ (the decidable fragment of FOL with no function symbols and only 2 variables [16]) theory. For example, the concept axiom $C \sqsubseteq D \sqcap \exists R.E$ can be translated into the following $\mathcal{L}^2$ axiom: $\forall x(\phi_C(x) \rightarrow \phi_D(x) \wedge \exists y(\phi_R(x, y) \wedge \phi_E(y)))$, where ϕ_C, ϕ_D, ϕ_E are unary predicates and ϕ_R is a binary predicate.

2.2 Possibilistic Logic

Possibilistic logic [5] is a weighted logic where each classical logic formula is associated with a number in $(0, 1]$. Semantically, the most basic and important notion is *possibility distribution* π: $\Omega \rightarrow [0, 1]$, where Ω is the set of all classical interpretations. $\pi(\omega)$ represents the degree of compatibility of interpretation ω with available beliefs. From *possibility distribution* π, two measures can be determined, one is the possibility degree of formula ϕ, defined as $\Pi(\phi) = max\{\pi(\omega) : \omega \models \phi\}$, the other is the necessity or certainty degree of formula ϕ, defined as $N(\phi) = 1 - \Pi(\neg\phi)$.

At syntactical level, a *possibilistic formula* is a pair (ϕ, α) consisting of a classical logic formula ϕ and a degree α expressing certainty or priority. A possibilistic knowledge base is the set of possibilistic formulas of the form $B = \{(\phi_i, \alpha_i) : i = 1, ..., n\}$. The classical base associated with B, denoted B^*, is defined as $B^* = \{\phi_i | (\phi_i, \alpha_i) \in B\}$. A possibilistic knowledge base is consistent iff its classical base is consistent.

Given a possibilistic knowledge base B and $\alpha \in (0, 1]$, the α-cut (strict α-cut) of B is $B_{\geq\alpha} = \{\phi \in B^* | (\phi, \beta) \in B \text{ and } \beta \geq \alpha\}$ ($B_{>\alpha} = \{\phi \in B^* | (\phi, \beta) \in B \text{ and } \beta > \alpha\}$). The *inconsistency degree* of B, denoted $Inc(B)$, is defined as $Inc(B) = max\{\alpha_i : B_{\geq\alpha_i} \text{ is inconsistent}\}$.

There are two possible definitions of inference in possibilistic logic.

Definition 1. *Let B be a possibilistic knowledge base.*

- *A formula ϕ is said to be a* plausible consequence *of B, denoted by $B \vdash_P \phi$, iff $B_{>Inc(B)} \vdash \phi$.*
- *A formula ϕ is said to be a* possibilistic consequence *of B to degree α, denoted by $B \vdash_\pi (\phi, \alpha)$, iff the following conditions hold: (1) $B_{\geq\alpha}$ is consistent, (2) $B_{\geq\alpha} \vdash \phi$, (3) $\forall \beta > \alpha$, $B_{\geq\beta} \nvdash \phi$.*

According to Definition 1, an inconsistent possibilistic knowledge base can nontrivially infer conclusion, so it is inconsistency tolerent. However, it suffers from the "drowning problem" [2]. That is, given an inconsistent possibilistic knowledge base B, formulas whose certainty degrees are not larger than $Inc(B)$ are completely useless for nontrivial deductions. For instance, let $B = \{(p, 0.9), (\neg p, 0.8), (r, 0.6), (q, 0.7)\}$, it is clear that B is equivalent to $B = \{(p, 0.9), (\neg p, 0.8)\}$ because $Inc(B) = 0.8$. So $(q, 0.7)$ and $(r, 0.6)$ are not used in the possibilistic inference.

Several variants of possibilistic inference have been proposed to avoid the drowning effect. One of them, called linear order inference, is defined as follows.

Definition 2. *Let $B = \{(\phi_i, \alpha_i) : i = 1, ..., n\}$ be a possibilistic knowledge base. Suppose β_j $(j = 1, ..., k)$ are all distinct weights appearing in B such that $\beta_1 > \beta_2 > ... > \beta_k$. Let $\Sigma_B = (S_1, ..., S_k)$, where $S_i = \{\phi_l : (\phi_l, \alpha_l) \in B, \alpha_l = \beta_i\}$, and $\Sigma_{LO,B} = \bigcup_{i=1}^{k} S_i^{'}$, where $S_i^{'}$ is defined by $S_i^{'} = S_i$ if $S_i \cup \bigcup_{j=1}^{i-1} S_j^{'}$ is consistent, $\emptyset$ otherwise. A formula ϕ is said to be a linear consequence of B, denoted $B \vdash_{LO} \phi$, iff $\Sigma_{LO,B} \vdash \phi$.*

The linear order approach does not stop at the inconsistency degree of possibilistic knowledge base B. It takes into account of formulas whose certainty degrees are less than the inconsistency degree.

3 Possibilistic Description Logics

3.1 Syntax

The syntax of possibilistic DL is based on the syntax of classical DL. A *possibilistic axiom* is a pair (ϕ, α) consisting of an axiom ϕ and a weight $\alpha \in (0, 1]$. A *possibilistic TBox* (resp., *ABox*) is a finite set of possibilistic axioms (ϕ, α), where ϕ is an TBox (resp., ABox) axoim. A possibilistic DL knowledge base $\mathcal{B} = (\mathcal{T}, \mathcal{A})$ consists of a possibilistic TBox $\mathcal{T}$ and a possibilistic ABox $\mathcal{A}$. We use $\mathcal{T}^*$ to denote the classical DL axioms associated with $\mathcal{T}$, i.e., $\mathcal{T}^* = \{\phi_i : (\phi_i, \alpha_i) \in \mathcal{T}\}$ ($\mathcal{A}^*$ can be defined similarly). The classical base $\mathcal{B}^*$ of a possibilistic DL knowledge base is $\mathcal{B}^* = (\mathcal{T}^*, \mathcal{A}^*)$. A possibilistic DL knowledge base $\mathcal{B}$ is inconsistent if and only if $\mathcal{B}^*$ is inconsistent.

Given a possibilistic DL knowledge base $\mathcal{B} = (\mathcal{T}, \mathcal{A})$ and $\alpha \in (0, 1]$, the α-cut of $\mathcal{T}$ is $\mathcal{T}_{\geq\alpha} = \{\phi \in \mathcal{B}^* | (\phi, \beta) \in \mathcal{T}$ *and* $\beta \geq \alpha\}$ (the α-cut of $\mathcal{A}$, denoted as $\mathcal{A}_{\geq\alpha}$, can be defined similarly). The strict α-cut of $\mathcal{T}$ (resp., $\mathcal{A}$) can be defined similarly as the strict cut in possibilistic logic. The α-cut (resp., strict α-cut) of $\mathcal{B}$ is $\mathcal{B}_{\geq\alpha} = (\mathcal{T}_{\geq\alpha}, \mathcal{A}_{\geq\alpha})$ (resp., $\mathcal{B}_{>\alpha} = (\mathcal{T}_{>\alpha}, \mathcal{A}_{>\alpha})$). The *inconsistency degree* of $\mathcal{B}$, denoted $Inc(\mathcal{B})$, is defined as $Inc(\mathcal{B}) = max\{\alpha_i : \mathcal{B}_{\geq\alpha_i}$ *is inconsistent*$\}$.

We use the following example as a running example throughout this paper.

Example 1. Suppose we have a possibilistic DL knowledge base $\mathcal{B} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T} = \{(Eat_{fish} \sqsubseteq Swim, 0.6), (Bird \sqsubseteq Fly, 0.8), (HasWing \sqsubseteq Bird,\ 0.95)\}$ and $\mathcal{A} = \{(Bird\ (chirpy), 1), (HasWing(tweety), 1), (\neg Fly(tweety), 1)\}$. The TBox $\mathcal{T}$ states that it is rather certain that birds can fly and it is almost certain that something with wing is a bird. The ABox $\mathcal{A}$ states that it is certain that tweety has wing and it cannot fly, and chirpy is a bird. Let $\alpha = 0.8$. We then have $\mathcal{B}_{\geq 0.8} = (\mathcal{T}_{\geq 0.8}, \mathcal{A}_{\geq 0.8})$, where $\mathcal{T}_{\geq 0.8} = \{Bird \sqsubseteq Fly, HasWing \sqsubseteq Bird\}$ and $\mathcal{A}_{\geq 0.8} = \{HasWing(tweety), \neg Fly(tweety), Bird(chirpy)\}$. It is clear that $\mathcal{B}_{\geq \alpha}$ is inconsistent. Now let $\alpha = 0.95$. Then $\mathcal{B}_{\geq \alpha} = (\mathcal{T}_{\geq 0.95}, \mathcal{A}_{\geq 0.95})$, where $\mathcal{T}_{\geq 0.95} = \{HasWing \sqsubseteq Bird\}$ and $\mathcal{A}_{\geq 0.95} = \{HasWing(tweety),\ \neg Fly(tweety), Bird\ (chirpy)\}$. So $\mathcal{B}_{\geq \alpha}$ is consistent. Therefore, $Inc(\mathcal{B}) = 0.8$.

3.2 Semantics

The semantics of possibilistic DL is defined by a *possibility distribution* π over the set $\mathbf{I}$ of all classical description logic interpretations, i.e., $\pi : \mathbf{I} \rightarrow [0, 1]$. $\pi(I)$ represents the degree of compatibility of interpretation I with available information. For two interpretations I_1 and I_2, $\pi(I_1) > \pi(I_2)$ means that I_1 is preferred to I_2 according to the available information. Given a possibility distribution π, we can define the possibility measure Π and necessity measure N as follows: $\Pi(\phi) = max\{\pi(I) : I \in \mathbf{I}, I \models \phi\}$ and $N(\phi) = 1 - max\{\pi(I) : I \not\models \phi\}$[2]. Unlike possibilistic logic, the necessary measure cannot be not defined by the possibility measure because the negation of an axiom is not defined in traditional DLs. However, given a DL axiom ϕ, let us define the negation of ϕ as $\neg\phi = \exists(C \sqcap \neg D)$ if $\phi = C \sqsubseteq D$ and $\neg\phi = \neg C(a)$ if $\phi = C(a)$, where $\exists(C \sqcap \neg D)$ is an existence axiom (see the discussion of negation of a DL axiom in [9]), then it is easy to check that $N(\phi) = 1 - \Pi(\neg\phi)$. Given two possibility distributions π and π', we say that π is more specific (or more informative) than π' iff $\pi(I) \leq \pi'(I)$ for all $I \in \Omega$. A possibility distribution π satisfies a possibilistic axiom (ϕ, α), denoted $\pi \models (\phi, \alpha)$, iff $N(\phi) \geq \alpha$. It satisfies a possibilistic DL knowledge base $\mathcal{B}$, denoted $\pi \models \mathcal{B}$, iff it satisfies all the possibilistic axioms in $\mathcal{B}$.

Given a possibilistic DL knowledge base $\mathcal{B} = \langle \mathcal{T}, \mathcal{A} \rangle$, we can define a possibility distribution from it as follows: for all $I \in \mathbf{I}$,

$$\pi_{\mathcal{B}}(I) = \begin{cases} 1 & \text{if } \forall \phi_i \in \mathcal{T}^* \cup \mathcal{A}^*, I \models \phi_i, \\ 1 - max\{\alpha_i | I \not\models \phi_i, (\phi_i, \alpha_i) \in \mathcal{T} \cup \mathcal{A}\} & \text{otherwise.} \end{cases} \quad (1)$$

As in possibilistic logic, we can also show that the possibility distribution defined by Equation 1 is the least specific possibility distribution satisfying $\mathcal{B}$. Let us consider Example 1 again. $I = \langle \Delta^I, \cdot^I \rangle$ is an interpretation, where $\Delta^I = \{tweety, chirpy\}$ and $Bird^I = \{tweety, chirpy\}$, $Fly^I = \{chirpy\}$, and $HasWing^I = \{tweety\}$. It is clear that I satisfies all the axioms except $Bird \sqsubseteq Fly$ (whose weight is 0.8), so $\pi_{\mathcal{B}}(I) = 0.2$.

[2] The definition of necessity measure is pointed out by one of the reviewers.

Let us give some properties of the possibility distribution defined by Equation (1).

Theorem 1. *Let $\mathcal{B}$ be a possibilistic DL knowledge base and $\pi_{\mathcal{B}}$ be the possibility distribution obtained by Equation (1). Then $\mathcal{B}$ is consistent if and only if there exists an interpretation $\mathcal{I}$ such that $\pi_{\mathcal{B}}(\mathcal{I}) = 1$.*

Proposition 1. *Let $\mathcal{B}$ be a possibilistic DL knowledge base and $\pi_{\mathcal{B}}$ be the possibility distribution obtained by Equation 1. Then $Inc(\mathcal{B}) = 1 - max_{I \in \mathbf{I}} \pi_{\mathcal{B}}(I)$.*

3.3 Possibilistic Inference in Possibilistic DLs

We consider the following inference services in possibilistic DLs.

- Instance checking: an individual a is a *plausible* instance of a concept C with respect to a possibilistic DL knowledge base $\mathcal{B}$, written $\mathcal{B} \models_P C(a)$, if $\mathcal{B}_{>Inc(\mathcal{B})} \models C(a)$.
- Instance checking with necessity degree: an individual a is an instance of a concept C to degree α with respect to $\mathcal{B}$, written $\mathcal{B} \models_\pi (C(a), \alpha)$, if the following conditions hold: (1) $\mathcal{B}_{\geq\alpha}$ is consistent, (2) $\mathcal{B}_{\geq\alpha} \models C(a)$, (3) for all $\beta{>}\alpha$, $\mathcal{B}_{\geq\beta} \not\models C(a)$.
- Instance checking with necessity degree: an individual a is an instance of a concept C to degree α with respect to $\mathcal{B}$, written $\mathcal{B} \models_\pi (C(a), \alpha)$, if the following conditions hold: (1) $\mathcal{B}_{\geq\alpha}$ is consistent, (2) $\mathcal{B}_{\geq\alpha} \models C(a)$, (3) for all $\beta{>}\alpha$, $\mathcal{B}_{\geq\beta} \not\models C(a)$.
- Subsumption with necessity degree: a concept C is subsumed by a concept D to a degree α with respect to a possibilistic DL knowledge base $\mathcal{B}$, written $\mathcal{B} \models_\pi (C \sqsubseteq D, \alpha)$, if the following conditions hold: (1) $\mathcal{B}_{\geq\alpha}$ is consistent, (2) $\mathcal{B}_{\geq\alpha} \models C \sqsubseteq D$, (3) for all $\beta{>}\alpha$, $\mathcal{B}_{\geq\beta} \not\models C \sqsubseteq D$.

We illustrate the inference services by reconsidering Example 1.

Example 2. (Example 1 continued) According to Example 1, we have $Inc(\mathcal{B}) = 0.8$ and $\mathcal{B}_{>0.8} = (\mathcal{T}_{>0.8}, \mathcal{A}_{>0.8})$, where $\mathcal{T}_{>0.8} = \{HasWing \sqsubseteq Bird\}$ and $\mathcal{A}_{>0.8} = \{HasWing(tweety), \neg Fly\,(tweety), Bird(chirpy)\}$. Since $\mathcal{B}_{>0.8} \models Bird(tweety)$, we can infer that *tweety* is plausible to be a bird from $\mathcal{B}$. Furthermore, since $\mathcal{B}_{\geq 0.95} \models Bird(tweety)$ and $\mathcal{B}_{\geq 1} \not\models Bird(tweety)$, we have $\mathcal{B} \models_\pi (Bird(tweety), 0.95)$. That is, we are almost certain that tweety is a bird.

3.4 Linear Order Inference in Possibilistic DLs

Possibilistic inference in possibilistic DL inherits the drowning effect of possibilistic inference in possibilistic logic. We adapt and generalize the linear order inference to deal with the drowning problem.

Definition 3. *Let $\mathcal{B} = (\mathcal{T}, \mathcal{A})$ be a possibilistic DL knowledge base. Suppose β_j $(j = 1, ..., k)$ are all distinct weights appearing in $\mathcal{B}$ such that $\beta_1 > \beta_2 > ... > \beta_k$. Let $\mathcal{B}' = \cup\mathcal{T} \cup \mathcal{A}$. Let $\Sigma_{\mathcal{B}} = (\mathcal{S}_1, ..., \mathcal{S}_k)$, where $\mathcal{S}_i = \{(\phi_l, \alpha_l) : (\phi_l, \alpha_l) \in \mathcal{B}', \alpha_l = \beta_i\}$, and $\Sigma_{LO,\mathcal{B}} = \bigcup_{i=1}^{k} \mathcal{S}_i^{'}$, where $\mathcal{S}_i^{'}$ is defined by $\mathcal{S}_i^{'} = \mathcal{S}_i$ if $\mathcal{S}_i \cup \bigcup_{j=1}^{i-1} \mathcal{S}_j^{'}$ is consistent, $\emptyset$ otherwise. Let ϕ be a query of the form $C(a)$ or $C \sqsubseteq D$. Then*

Algorithm 1. Compute the inconsistency degree

Data: $\mathcal{B} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T} \cup \mathcal{A} = \{(\phi_i, \alpha_i) : \alpha_i \in (0, 1], i = 1, ..., n\}$, where n is the number of axioms in the testing ontology $\mathcal{B}$;
Result: The inconsistency degree d
begin
  $b := 0$ // b is the begin pointer of the binary search
  $m := 0$ // m is the middle pointer of the binary search
  $d := 0.0$ // The initial value of inconsistency degree d is set to be 0.0
  $W = Asc(\alpha_1, ..., \alpha_n)$
  $W(-1) = 0.0$ // The special element -1 of W is set to be 0.0
  $e := |W| - 1$ // e is the end pointer of the binary search
  if $\mathcal{B}_{\geq W(0)}$ *is consistent* **then**
    d:=0.0
  else
    while $b \leq e$ **do**
      if $b = e$ **then**
        return b
      $m := \ulcorner (b + e)/2 \urcorner$
      if $\mathcal{B}_{\geq W(m)}$ *is consistent* **then**
        $e := m - 1$
      else
        $b := m + 1$
    $d := W(b)$
end

- ϕ *is said to be a consequence of* $\mathcal{B}$ *w.r.t the linear order policy, denoted* $\mathcal{B} \vdash_{LO} \phi$*, iff* $(\Sigma_{LO,\mathcal{B}})^* \vdash \phi$.
- ϕ *is said to be a weighted consequence of* $\mathcal{B}$ *to a degree* α *w.r.t the linear order policy, denoted* $\mathcal{B} \vdash_{LO} (\phi, \alpha)$*, iff* $\Sigma_{LO,\mathcal{B}} \vdash_{\pi} (\phi, \alpha)$.

In Definition 3, we not only define the consequence of a possibilistic DL knowledge base *w.r.t* the linear order policy, but also the weighted consequence of it. The weighted consequence of $\mathcal{B}$ is based on the possibilistic inference.

Example 3. (Example 1 continued) Let $\phi = Eat_{fish} \sqsubseteq Swim$. According to Example 2, ϕ is not a consequence of $\mathcal{B}$ *w.r.t.* the possibilistic inference. Since $\Sigma_{\mathcal{B}} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4)$, where $\mathcal{S}_1 = \mathcal{A}$, $\mathcal{S}_2 = \{(HasWing \sqsubseteq Bird, 0.95)\}$, $\mathcal{S}_3 = \{(Bird \sqsubseteq Fly, 0.8)\}$ and $\mathcal{S}_4 = \{(Eat_{fish} \sqsubseteq Swim, 0.6)\}$, we have $\Sigma_{LO,\mathcal{B}} = S_1 \cup S_2 \cup S_4$. It is easy to check that $\mathcal{B} \vdash_{LO} (Eat_{fish} \sqsubseteq Swim, 0.6)$.

4 Algorithms for Inference in Possibilistic DLs

We give algorithms for the inference in possibilistic DLs.

Algorithm 1 computes the inconsistency degree of a possibilistic DL knowledge base using a binary search. The function *Asc* takes a finite set of numbers in $(0, 1]$ as input and returns a vector which contains those distinct numbers in the

Algorithm 2. Possibilistic inference with certainty degrees

Data: $\mathcal{B} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T} \cup \mathcal{A} = \{(\phi_i, \alpha_i) : \alpha_i \in (0, 1], i = 1, ..., n\}$; a DL axiom ϕ.
Result: The certainty degree w associated with a query ϕ
begin
  $m := 0$
  $w := 0.0$ // The initial certainty degree of ϕ is set to be 0.0
  $W = Asc(\alpha_1, ..., \alpha_n)$
  $W(-1) = 0.0$
  $e := |W| - 1$
  compute l such that $W(l) = Inc(\mathcal{B})$ //$Inc(\mathcal{B})$ is computed by Algorithm 1
  $b := l + 1$
  if $\mathcal{B}_{\geq W(b)} \models \phi$ **then**
    while $b \leq e$ **do**
      if $b = e$ **then**
        return b
      $m := \ulcorner (b + e)/2 \urcorner$
      if $\mathcal{B}_{\geq W(m)} \not\models \phi$ **then**
        $e := m - 1$
      else
        $b := m + 1$
    $w := W(b)$
end

set in an ascending order. For example, $Asc(0.2, 0.3, 0.3, 0.1) = (0.1, 0.2, 0.3)$. Let $W = (\beta_1, ..., \beta_n)$ is a vector consisting of n distinct numbers, then $W(i)$ denotes β_i. If the returned inconsistency degree is 0, that is $W(-1) = 0$, it shows the ontology to be queried is consistent.

Since Algorithm 1 is based on binary search, to compute the inconsistency degree, it is easy to check that the algorithm requires at most $\ulcorner log_2 n \urcorner + 1$ satisfiability check using a DL reasoner.

Algorithm 2 returns the necessity degree of an axiom inferred from a possibilistic DL knowledge base *w.r.t* the possibilistic inference. We compute the inconsistency degree of the input ontology. If the axiom is a plausible consequence of a possibilistic DL knowledge base, then we compute its necessity degree using a binary search (see the first "if" condition). Otherwise, its necessity degree is 0, i.e., the default value given to w. Note that our algorithm is different from the algorithm given in [15] for computing the necessity of a formula in possibilistic logic (this algorithm needs to compute the negation of a formula, which is computationally hard in DLs according to [9]). We consider only subsumption checking here. However, the algorithm can be easily extended to reduce instance checking as well.

In Algorithm 3, we call Algorithm 1 and Algorithm 2 to compute the certainty degree of the query ϕ *w.r.t* the linear order inference. In the "while" loop, the first "if" condition checks if the inconsistency degree is greater than 0 and then

Algorithm 3. Linear order inference with certainty degrees

Data: $\mathcal{B} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T} \cup \mathcal{A} = \{(\phi_i, \alpha_i) : \alpha_i \in (0, 1], i = 1, ..., n\}$; a DL axiom ϕ.
Result: The certainty degree w associated with a query ϕ
begin
  $d := 0.0$ // The initial inconsistency degree is set to be 0.0
  $w := 0.0$ // The initial certainty degree of ϕ is set to be 0.0
  $hasAnswer := false$
  $W = Asc(\alpha_1, ..., \alpha_n)$
  $e := |W| - 1$ // e is a global variable to pass values to the subroutines
  while $!hasAnswer$ **do**
    if $d > 0$ **then**
      $e := d - 1$
      $\mathcal{B} := \mathcal{B} \setminus \mathcal{B}_{=d}$
      $W := W \setminus d$
    $d := alg1(\mathcal{B})$, where $alg1$ is Algorithm 1
    if $\mathcal{B}_{>d} \models \phi$ **then**
      $hasAnswer := true$
    if $d \leq 0$ **then**
      break
  if $hasAnswer$ **then**
    $w := alg2(\mathcal{B}, \phi)$, where $alg2$ is Algorithm 2
end

delete the axioms whose necessity degrees are equal to the inconsistency degree. After that, we call Algorithm 1 to compute the inconsistency degree of the initial knowledge base or knowledge base obtained from the first "if" loop. Then the second "if" condition checks if the axiom is a plausible consequence of the possibilistic DL knowledge base and end the "while" loop if the answer is positive. The final "if" condition simply tests if the possibilistic DL knowledge base is consistent or not and terminate the "while" loop if the answer is positive. Finally, we compute the certainty degree of ϕ by calling Algorithm 2. This algorithm need to call polynomial times of satisfiability check using a DL reasoner.

Algorithms 2 and 3 compute inference with certainty degree because it is more difficult to obtain the certainty degree of an inferred axiom. They can be easily revised to compute plausible consequence. Because of the page limit, we do not provide the details here.

Proposition 2. *Let $\mathcal{B}$ be a possibilistic DL knowledge base and ϕ be a DL axiom. Deciding whether $\mathcal{B} \models_P \phi$ requires $\lceil log_2 n \rceil + 1$ satisfiability check using a DL reasoner, where n is the number of distinct certainty degrees in $\mathcal{B}$. Furthermore, deciding whether $\mathcal{B} \models_\pi (\phi, \alpha)$ requires at most $\lceil log_2 n \rceil + \lceil log_2 n - l \rceil + 1$ satisfiability check using a DL reasoner, where where n is the number of distinct certainty degrees in $\mathcal{B}$ and l is the inconsistency degree of $\mathcal{B}$.*

5 Implementation and Results

To test our algorithms, we have implemented them in Java using KAON2[3]. All tests were performed on a laptop computer with a 1.7GHz Intel processor, 1 GB of RAM, running Windows XP Service Pack 2. Sun's Java 1.5.0 Update 6 is used, and the virtual memory of the Java virtual machine was limited to 800M.

5.1 Results

We use ontologies miniTambis[4] and *proton*_100_*all*[5] as test data. The first ontology contains more than 170 concepts, 35 properties, 172 axioms and 30 unsatisfiable concepts. The second ontology has 175 concepts, 266 properties, 3 unsatisfiable concepts and about 1100 axioms. Both ontologies are consistent but contain some unsatisfiable concepts. We added some instances to the unsatisfiable concepts to make the ontology inconsistent. We get possibilistic DL knowledge bases from miniTambis and *proton*_100_*all* by randomly attaching certainty degrees to them and using a separate ontology to store the information on the certain degrees. Given a set of certainty degrees $W = (w_1, w_2, ..., w_n),\ w_i \in (0, 1],\ i = 1, ..., n$, an automatic mechanism is applied to randomly choose a certainty degree w_i for each axiom in the ontology to be queried.

In Table 2, some results based on the two ontologies above are given, where $|W|$ means the number of different certainty degrees for testing. The rows corresponding to Algorithm 2 and Algorithm 3 describe the time spending on a specific reasoning task which is instance checking, i.e., the third row shows the time spent by executing Algorithm 2 and the last row is for Algorithm 3. For each column in an ontology, we randomly attach the certainty degrees to axioms in the ontology and give the time spending on a specific reasoning task. Therefore, different columns gives results for different possibilistic DL knowledge bases which may generate from the same ontology.

According to the table, in some cases, the time spent on query by Algorithm 2 and Algorithm 3 is almost the same (see columns 1 and 2 for miniTambis). For example, when the axiom ϕ to be queried can be inferred by $\mathcal{B}_{Inc(\mathcal{B})}$. In other cases, it takes much more time for Algorithm 3 to return the result than Algorithm 2. For example, see columns 3 for miniTambis, it takes 2 seconds to

Table 2. The results from Algorithm 2 and Algorithm 3

Ontology	*miniTambis*								*proton*_100_*all*							
$\|W\|$	10				30				10				30			
Algorithm 2 (s)	6	8	2	10	16	9	13	8	5	11	5	6	9	12	8	13
Algorithm 3 (s)	6	9	12	23	16	8	33	44	5	10	15	12	8	12	19	24

[3] http://kaon2.semanticweb.org/
[4] http://www.mindswap.org/2005/debugging/ontologies/
[5] http://wasp.cs.vu.nl/knowledgeweb/d2163/learning.html

get result from Algorithm 2 and 12 second from Algorithm 3. This is because Algorithm 2 stops when $\mathcal{B}_{\geq W(d)} \models \phi$ is not satisfied. But for Algorithm 3, it will not stop until $\mathcal{B}_{\geq W(d)} \models \phi$ is satisfied, or no more inconsistency degree can be found.

6 Related Work

Our work differs from existing work on extending description logics by possibilistic logic in following points: (1) we provided semantics of the possibilistic description logic, (2) we considered two inference services and give algorithms for computing the consequences of the inference, (3) we proposed a linear order inference which is a drowning-free variant of possibilistic inference and provided algorithm for it, (4) we implemented the proposed algorithm and provided for evaluation results.

Other approaches that extend description logics with uncertainty reasoning are probabilistic description logics [12,10] and fuzzy extension of description logics (e.g., [18,17]). The main difference between possibilistic extension and probabilistic extension lies in the fact that possibilistic logic is a qualitative representation of uncertainty, whilst probabilistic extension is on quantitative aspects of uncertainty. Furthermore, possibilistic DLs can be used to deal with inconsistency and probabilistic DLs are not used for this purpose. Arguably, fuzzy description logics can be used to deal with uncertainty. In possibilistic DLs, the truth value of an axiom is still two-valued, whilst in fuzzy DLs, the truth value of an axiom is multi-valued.

7 Conclusions and Future Work

We gave a possibilistic extension of description logics in this paper. We first defined syntax and semantics of possibilistic description logics. Then we consider inference problems in our logics: possibilistic inference and linear order inference. Algorithms were given to check the inference and we implemented the algorithms. As far as we know, this is the first work which discusses how to implement possibilistic description logics. Finally, we report some preliminary but encouraging experimental results.

The algorithms for possibilistic inference of our logics proposed in this paper is independent of DL reasoner. In our future work, we plan to give more efficient reasoning approaches by generalizing the resolution-based reasoning approach for KAON2. Another future work is that we may interpret the concept axioms by possibilistic conditioning and explore the nonmonotonic feature of possibilistic description logics.

Acknowledgements. Thank the anonymous referees for their helpful comments. Research presented in this paper was partially supported by the European Commission NeOn (IST-2006-027595, http://www.neon-project.org/), the Knowledge Web projects (IST-2004-507842, http://knowledgeweb.semanticweb. org/), and the

X-Media project (www.x-media-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-026978.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): Description Logic Handbook: Theory, implementation and applications. Cambridge University Press, Cambridge (2003)
2. Benferhat, S., Cayrol, C., Dubois, D., Lang, J., Prade, H.: Inconsistency management and prioritized syntax-based entailment. In: Proc. of IJCAI'93, pp. 640–647. Morgan Kaufmann, San Francisco (1993)
3. Benferhat, S., Dubois, D., Prade, H.: Representing default rules in possibilistic logic. In: Proc. of KR'92, pp. 673–684 (1992)
4. Benferhat, S., Lagrue, S., Papini, O.: Reasoning with partially ordered information in a possibilistic logic framework. Fuzzy Sets and Systems 144(1), 25–41 (2004)
5. Dubois, D., Lang, J., Prade, H.: Possibilistic logic. In: Handbook of Logic in Artificial Intelligence and Logic Programming, pp. 439–513. Oxford University Press, Oxford (1994)
6. Dubois, D., Mengin, J., Prade, H.: Possibilistic uncertainty and fuzzy features in description logic: A preliminary discussion. In: Capturing Intelligence: Fuzzy Logic and the Semantic WEb, pp. 101–113. Elsevier, Amsterdam (2006)
7. Dubois, D., Prade, H.: Epistemic entrenchment and possibilistic logic. Artif. Intell. 50(2), 223–239 (1991)
8. Dubois, D., Prade, H.: Possibility theory: qualitative and quantitative aspects. In: Handbook of Defeasible Reasoning and Uncertainty Management Systems, pp. 169–226 (1998)
9. Flouris, G., Huang, Z., Pan, J.Z., Plexousakis, D., Wache, H.: Inconsistencies, negations and changes in ontologies. In: Proc. of AAAI'06 (2006)
10. Giugno, R., Lukasiewicz, T.: P-shoq(d): A probabilistic extension of shoq(d) for probabilistic ontologies in the semantic web. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, pp. 86–97. Springer, Heidelberg (2002)
11. Haase, P., Völker, J.: Ontology learning and reasoning - dealing with uncertainty and inconsistency. In: Proc. of URSW'05, pp. 45–55 (2005)
12. Heinsohn, J.: Probabilistic description logics. In: Proc. of UAI'94, pp. 311–318 (1994)
13. Hollunder, B.: An alternative proof method for possibilistic logic and its application to terminological logics. Int. J. Approx. Reasoning 12(2), 85–109 (1995)
14. Horrocks, I., Sattler, U.: A tableaux decision procedure for SHOIQ. In: Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005) (2005)
15. Lang, J.: Possibilistic logic: complexity and algorithms. In: Handbook of Defeasible Reasoning and Uncertainty Management Systems, pp. 179–220 (2000)
16. Mortimer, M.: On languages with two variables. Zeitschr. f. math. Logik u. Grundlagen d. Math. 21, 135–140 (1975)
17. Stoilos, G., Stamou, G., Pan, J.Z., Tzouvaras, V., Horrocks, I.: Reasoning with very expressive fuzzy description logics. J. Artif. Intell. Res. (2007)
18. Straccia, U.: Reasoning within fuzzy description logics. J. Artif. Intell. Res. 14, 137–166 (2001)

Information Affinity: A New Similarity Measure for Possibilistic Uncertain Information

Ilyes Jenhani[1], Nahla Ben Amor[1], Zied Elouedi[1], Salem Benferhat[2], and Khaled Mellouli[1]

[1] LARODEC, Institut Supérieur de Gestion de Tunis, Tunisia
[2] CRIL, Université d'Artois, Lens, France
ilyes.j@lycos.com, nahla.benamor@gmx.fr, zied.elouedi@gmx.fr, benferhat@cril.univ-artois.fr, khaled.mellouli@ihec.rnu.tn

Abstract. This paper addresses the issue of measuring similarity between pieces of uncertain information in the framework of possibility theory. In a first part, natural properties of such functions are proposed and a survey of the few existing measures is presented. Then, a new measure so-called Information Affinity is proposed to overcome the limits of the existing ones. The proposed function is based on two measures, namely, a classical informative distance, e.g. Manhattan distance which evaluates the difference, degree by degree, between two normalized possibility distributions and the well known inconsistency measure which assesses the conflict between the two possibility distributions. Some potential applications of the proposed measure are also mentioned in this paper.

Keywords: Possibility theory, Similarity, Divergence measure, Distance, Inconsistency measure.

1 Introduction

Most of real-world decision problems are faced with uncertainty. Uncertainty about values of given variables (e.g. the type of a detected target in military applications, the disease affecting a patient in medical applications, etc.) can result from some errors and hence from non-reliability (in the case of sensors) or from different background knowledge (in the case of agents: doctors, etc.). As a consequence, it is possible to obtain different uncertain pieces of information about a given value from different sources. Obviously, comparing these pieces of information could be very interesting to support decision making.

Comparing pieces of uncertain information given by several sources has attracted a lot of attention for a long time. For instance, we can mention the well-known Euclidean and KL-divergence [17] for comparing probability distributions. Another distance has been proposed by Chan and al. [4] for bounding probabilistic belief change. Moving to belief function theory [20], several distance measures between bodies of evidence deserve to be mentioned. Some distances

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 840–852, 2007.

have been proposed as measures of performance (MOP) of identification algorithms [8] [14]. Another distance was used for the optimization of the parameters of a belief k-nearest neighbor classifier [26]. In [21], the authors proposed a distance for the quantification of errors resulting from basic probability assignment approximations. Similarity measures between two fuzzy sets A and B have been also proposed in the literature [6] [9] [23] [24]. For instance, in the work by Bouchon-Meunier and al. [3], the authors proposed a similarity measure between fuzzy sets as an extension of Tversky's model on crisp sets [22]. The measure was then used to develop an image search engine.

Contrary to probability, belief function and fuzzy set theories, few works are dedicated to distance measures in possibility theory despite its popularity. Hence, in this paper, we will focus on measures for the comparison of uncertain information represented by possibility distributions. In a first part, we will study the few existing works and show their limits, then, we will propose a new similarity measure, so-called *Information Affinity* which satisfies very natural properties. Our measure would be useful in many real-world applications where the uncertainty is modeled by means of possibility theory. For instance, it could be used as a critical parameter for distance based possibilistic machine learning algorithms, it could also be used for the evaluation of possibilistic classifiers, for the comparison of expert opinions, etc.

The rest of the paper is organized as follows: Section 2 starts by giving the necessary background concerning possibility theory. Section 3 provides different properties that a similarity measure should satisfy. Section 4 represents an overview of the existing similarity measures within the possibilistic setting with detailed examples and critics. The definition and the contrast of the new *Information Affinity* measure with existing measures are proposed in Section 5. Some potential applications of the proposed measure are shown in Section 6. Finally, Section 7 concludes the paper.

2 Possibility Theory

Possibility theory represents a non-classical theory (distinct from probability theory), first introduced by Zadeh [25] and then developed by several authors (e.g., Dubois and Prade [7]). In this section, we will give a brief recalling on possibility theory.

Possibility distribution

Given a universe of discourse $\Omega = \{\omega_1,\ \omega_2,\ ...,\ \omega_n\}$, a fundamental concept of possibility theory is the *possibility distribution* denoted by π. π corresponds to a function which associates to each element ω_i from the universe of discourse Ω a value from a bounded and linearly ordered valuation set (L,<). This value is called a *possibility degree*: it encodes our knowledge on the real world. Note that, in possibility theory, the scale can be numerical (e.g. L=[0,1]): in this case we have numerical possibility degrees from the interval [0,1] and hence we are dealing with the quantitative setting of the theory. In the qualitative setting, it is the ordering between the different possible values that is important.

By convention, $\pi(\omega_i) = 1$ means that it is fully possible that ω_i is the real world, $\pi(\omega_i) = 0$ means that ω_i cannot be the real world (is impossible). Flexibility is modeled by allowing to give a possibility degree from]0,1[. In possibility theory, extreme cases of knowledge are given by:

- *Complete knowledge*: $\exists \omega_i,\ \pi(\omega_i) = 1$ *and* $\forall\ \omega_j \neq \omega_i,\ \pi(\omega_j) = 0$.
- *Total ignorance*: $\forall\ \omega_i \in \Omega,\ \pi(\omega_i) = 1$ (all values in Ω are possible).

Possibility and Necessity measures

A Possibility measure is one of the fundamental concepts in possibility theory. From a possibility distribution, two dual measures can be derived: *Possibility* and *Necessity* measures. Given a possibility distribution π on the universe of discourse Ω, the corresponding possibility and necessity measures of any event $A \subseteq 2^{\Omega}$ are, respectively, determined by the formulas: $\Pi(A) = \max_{\omega \in A}\ \pi(\omega)$ and $N(A) = \min_{\omega \notin A}\ (1 - \pi(\omega)) = 1 - \Pi(\overline{A})$. $\Pi(A)$ evaluates at which level A is *consistent* with our knowledge represented by π while $N(A)$ evaluates at which level A is *certainly* implied by our knowledge represented by π.

Normalization

A possibility distribution π is said to be *normalized* if there exists at least one state $\omega_i \in \Omega$ which is totally possible (i.e. $\max_{\omega \in \Omega}\{\pi(\omega)\} = \pi(\omega_i)$=1). In the case of sub-normalized π,

$$Inc(\pi) = 1 - \max_{\omega \in \Omega}\{\pi(\omega)\} \tag{1}$$

is called the *inconsistency degree* of π. It is clear that, for normalized π, $\max_{\omega \in \Omega}\{\pi(\omega)\} = 1$, hence Inc($\pi$)=0. The measure Inc is very useful in assessing the degree of conflict between two distributions π_1 and π_2 which is given by $Inc(\pi_1 \wedge \pi_2)$. We take the $\wedge$ as the minimum operator. Obviously, when $\pi_1 \wedge \pi_2$ gives a sub-normalized possibility distribution, it indicates that there is a conflict between π_1 and π_2 ($Inc(\pi_1 \wedge \pi_2) \in]0, 1]$). On the other hand, when $\pi_1 \wedge \pi_2$ is normalized, there is no conflict and hence $Inc(\pi_1 \wedge \pi_2) = 0$.

Non-specificity

The degree of information uncertainty of a possibility distribution is called non-specificity and it can be measured by the so-called *U-uncertainty* criterion [10]. Given a permutation of the degrees of a possibility distribution $\pi = \langle \pi_{(1)},\ \pi_{(2)},\ ...,\ \pi_{(n)} \rangle$ such that $\pi_{(1)} \geq \pi_{(2)} \geq ... \geq \pi_{(n)}$, the *U-uncertainty* of π, is given by the formula:

$$U(\pi) = \sum_{i=2}^{n} (\pi_{(i)} - \pi_{(i+1)})\ log_2\ i\ +\ (1 - \pi_{(1)})\ log_2\ n \tag{2}$$

where $\pi_{(n+1)} = 0$ by convention [11]. Note that the range of U is [0, $log_2\ n$]. $U(\pi) = 0$ is obtained for the case of complete knowledge (no uncertainty) and $U(\pi) = log_2\ n$ is reached for instance in the case of total ignorance. Note also that the second term of the equation, i.e., $(1 - \pi_{(1)})\ log_2\ n$ generalizes U for sub-normalized π.

For the sake of simplicity, for the rest of the paper, a possibility distribution π on a finite set $\Omega = \{\omega_1, \omega_2, ..., \omega_n\}$ will be denoted by $\pi[\pi(\omega_1), \pi(\omega_2), ..., \pi(\omega_n)]$.

3 Natural Properties for a Possibilistic Similarity Measure

Let π_1 and π_2 be two *normalized* possibility distributions on the same universe of discourse Ω. A similarity measure is any function $s(\pi_1, \pi_2)$ satisfying the following six properties:

Property 1. Non-negativity $s(\pi_1, \pi_2) \geq 0$.
Property 2. Symmetry $s(\pi_1, \pi_2) = s(\pi_2, \pi_1)$.
Property 3. Upper bound and Non-degeneracy
If the range of s is the interval [0,1], then the upper bound of s is equal to 1. Formally, $\forall\ \pi_i,\ s(\pi_i, \pi_i) = 1$. So, s is maximal iff the arguments of s are identical.
Property 4. Lower bound
The lower bound of s is equal to 0. $s(\pi_1, \pi_2) = 0$ should be obtained only when we have to compare maximally contradictory possibility distributions. More formally, $s(\pi_1, \pi_2) = 0$ holds iff $\forall \omega_i \in \Omega$,
i) $\pi_1(\omega_i) \in \{0, 1\}$ and $\pi_2(\omega_i) \in \{0, 1\}$,
ii) *and* $\pi_2(\omega_i) = 1 - \pi_1(\omega_i)$
Item *i*) means that π_1 and π_2 should be binary. Now, since in this paper we only deal with normalized possibility distributions, items *i*) and *ii*) imply:
iii) $\exists\ \omega_q \in \Omega$ *s.t.* $\pi_1(\omega_q) = 1$
iv) $\exists\ \omega_p \in \Omega$ *s.t.* $\pi_1(\omega_p) = 0$

The following example illustrates properties (3) and (4):

Example 1. *Let X be a variable with an unknown value and let $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$ be the set representing the possible values of X. Let us take the possibility distribution representing the total knowledge on X given by an agent, i.e., $\pi_1[1, 0, 0, 0]$.*

Thus, the most similar possibility distribution to π_1 according to Property 3 is $\pi_2[1, 0, 0, 0]$ (the best case) and according to Property 4, the least similar possibility distribution to π_1 is $\pi_3[0, 1, 1, 1]$ (the worst case). In the above example, intuitively, the opinion of an agent who totally ignores the value of X, (i.e, $\pi_4[1, 1, 1, 1]$) is closer to π_1 than π_3. In fact, π_4 is in agreement with π_1 that ω_1 is fully possible but not for the remaining ω_i, on the other hand, π_3 does not agree with π_1 on any ω_i. Moreover, since π_4 is fully consistent with π_1 ($Inc(\pi_1 \wedge \pi_4) = 0$), it will be considered more close to π_1 than other possibility distribution π_j satisfying $Inc(\pi_1 \wedge \pi_j) = 1$.

Finally, in the case of equally inconsistent opinions with π_1, e.g., $\pi_5[0, 1, 1, 0]$ and $\pi_6[0, 1, 0, 0]$, the distribution having the smallest simple difference (degree by degree) will be considered the closest to π_1. Hence, π_6 is closer to π_1 than π_5. In fact, both π_5 and π_6 completely disagree with π_1 that ω_1 is the value of X, but π_6 agrees with π_1 that ω_3 and ω_4 are impossible while π_5 considers ω_3 as possible.

Property 5. Inclusion
If $\forall \omega_i \in \Omega,\ \pi_1(\omega_i) \leq \pi_2(\omega_i)$ and $\pi_2(\omega_i) \leq \pi_3(\omega_i)$, which by definition means that π_1 is more specific than π_2 which is in turn more specific than π_3, we obtain: $s(\pi_1, \pi_2) \geq s(\pi_1, \pi_3)$.

Property 6. Permutation
Suppose we have four possibility distributions π_1, π_2, π_3 and π_4 such that $s(\pi_1, \pi_2) > s(\pi_3, \pi_4)$. Suppose that $\forall j = 1..4$, *and* $\omega_p, \omega_q \in \Omega$, we have $\pi'_j(\omega_p) = \pi_j(\omega_q)$ and $\pi'_j(\omega_q) = \pi_j(\omega_p)$, hence we should obtain $s(\pi'_1, \pi'_2) > s(\pi'_3, \pi'_4)$.

4 Measuring Similarity of Possibilistic Uncertain Information

Measuring similarity of uncertainty based information has attracted a lot of attention in probability theory [4,17], in belief function theory [8,14,21,26], in fuzzy set theory [3,6,9,23,24] and in credal set theory [1]. This is not the case for possibilistic uncertain information, in fact, few works have been done in this direction. Let us present, chronologically, some of these measures and show their weaknesses in expressing information divergence between any given two agents (or sensors) who are expressing their opinions (or measures), especially, in the form of possibility distributions.

4.1 Information Closeness

The first paper, especially dedicated to the problem of measuring information similarity between two possibility distributions was the one of Higashi and Klir in 1983 [11]. They proposed an information variation based measure which they called *information closeness* denoted by G. Function G is computed using their U-uncertainty measure [10] (Equation (2)) and it is applicable to any pair of normalized possibility distributions. The less the value of G is, the more the information are similar (G behaves as a distance measure).

Definition 1. *Let π_1 and π_2 be two possibility distributions on the same universe of discourse Ω. The information closeness G between π_1 and π_2 is defined as:*

$$G(\pi_1, \pi_2) = g(\pi_1, \pi_1 \vee \pi_2) + g(\pi_2, \pi_1 \vee \pi_2) \tag{3}$$

where $g(\pi_i, \pi_j) = U(\pi_j) - U(\pi_i)$. $\vee$ is taken as the maximum operator and U is the non-specificity measure given by Equation (2). Consequently, function G can be written as $G(\pi_1, \pi_2) = 2 * U(\pi_1 \vee \pi_2) - U(\pi_1) - U(\pi_2)$.

Example 2. *Consider the following distributions π_1, π_2, π_3 and π_4 over $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$: π_1[1, 0.5, 0.3, 0.7], π_2[1, 0, 0, 0], π_3[0.9, 1, 0.3, 0.7], π_4[0, 1, 0.3, 0.7]. Let us try to find an order expressing which from the information given by π_2, π_3 and π_4 is closer to π_1. $G(\pi_1, \pi_2) = 1.12$, $G(\pi_1, \pi_3) = 0.52$, $G(\pi_1, \pi_4) = 1.08$. According to G, π_3 is the closest to π_1 and π_4 is closer to π_1 than π_2.*

The dissimilarity measure G does not satisfy *Property 4*. In fact, $G(\pi_i, \pi_j)$ should take its maximum value $\forall\ \pi_i,\ \pi_j$ satisfying items i) to iv) (see *Property 4*).

Example 3. *Let us consider these distributions:* $\pi_1[1,\ 0,\ 0,\ 0]$, $\pi_2[0,\ 1,\ 1,\ 1]$, $\pi_3[0,\ 1,\ 0,\ 1]$ *and* $\pi_4[1,\ 0,\ 1,\ 0]$ *. Clearly,* $\pi_1 = 1 - \pi_2$ *and* $\pi_3 = 1 - \pi_4$*. Hence, G should take its maximum value when comparing* π_1 *and* π_2 *as well as* π_3 *and* π_4*. Nevertheless, according to G, we obtain:* $G(\pi_1, \pi_2) = 2 * log_2(4) - log_2(3) = 2.41$, $G(\pi_3, \pi_4) = 2 * log_2(4) - 2 * log_2(2) = 2$*. It means that* π_3 *and* π_4 *are more similar to each others than* π_1 *and* π_2 *are, which is contrary to what we expect:* $G(\pi_1, \pi_2)$ *should be maximal and equal to* $G(\pi_3, \pi_4)$*.*

4.2 Sangüesa et al. Distance

In a work by Sangüesa et al. [19] focusing on learning possibilistic causal networks, the authors proposed a modified version of a distance measure [18] between two possibility distributions for DAG (Directed Acyclic Graph) learning and evaluation. This is done by measuring the distance (which must be minimized) between the possibility distribution implied by a DAG and the one underlying the database. This idea is based on the interpretation of independence as information similarity.

Definition 2. *Given two possibility distributions* π_1 *and* π_2 *on the same universe of discourse* Ω*. The distance between* π_1 *and* π_2 *is defined as the non-specificity of the distribution difference*

$$distance(\pi_1, \pi_2) = U(|\pi_1 - \pi_2|) \tag{4}$$

This measure gives different results from the previous one.

Example 4. *If we take the same distributions* π_1*,* π_2*,* π_3 *and* π_4 *of Example 2, we obtain:* $distance(\pi_1, \pi_2) = U([0, 0.5, 0.3, 0.7]) = 1.27$*,* $distance(\pi_1, \pi_3) = U([0.1, 0.5, 0, 0]) = 1.1$*,* $distance(\pi_1, \pi_4) = U([1, 0.5, 0, 0]) = 0.5$*. Hence according to this measure,* π_2 *remains the farthest but* π_4 *becomes the closest to* π_1*.*

This measure has a serious problem when the distribution difference ($|\pi_1 - \pi_2|$) is sub-normalized (which occurs most of the time). Indeed, it is in this situation that the second term of Equation (2) will be considered. If we concentrate in Equation (2), we can notice that measuring the non-specificity of a sub-normalized distribution π comes down to measure the non-specificity of its normalized distribution π' s.t $\pi'(\omega_i) = \pi(\omega_i) + 1 - max_{\omega \in \Omega}\{\pi(\omega)\}$. Obviously, this normalization scheme is not suited for the proposed distance. The following example shows this weakness:

Example 5. *Let us consider the following three possibility distributions:* $\pi_1[1,\ 0,\ 0,\ 0]$, $\pi_2[1,\ 0,\ 0,\ 0]$, $\pi_3[0,\ 1,\ 1,\ 1]$, $\pi_4[1,\ 1,\ 0,\ 0]$*. Clearly,* π_2 *is the closest possible distribution to* π_1 *(the best case) while* π_3 *is the farthest distribution (the worst case). Nevertheless, the distance measure does not agree:*
$distance(\pi_1, \pi_2) = U([0, 0, 0, 0]) = 2\ (maximum)$
$distance(\pi_1, \pi_3) = U([1, 1, 1, 1]) = 2\ (maximum)$

distance$(\pi_1, \pi_4) = U([0, 1, 0, 0]) = 0$ (*minimum*)
Hence, π_1 *and* π_2 *are maximally distant from each other which violates Property 4. Property 3 is also violated since, according to the example,* π_1 *and* π_4 *are maximally similar to each other.*

4.3 Information Divergence

A possibilistic analogy to the probabilistic measure of divergence was proposed by Kroupa [16]. The author has used the Choquet integral [5] as an aggregation operator of the possibility degrees characterizing the, generally, sub-normalized distribution difference ($\pi_d = |\pi_1(\omega_i) - \pi_2(\omega_i)|$, i=1..n) of any two normal distributions π_1 and π_2.

Definition 3. *Given two possibility distributions* π_1 *and* π_2 *on the same universe of discourse* Ω*, the measure of divergence* $D(\pi_1|\pi_2)$ *is defined as the discrete Choquet integral of the degrees of* π_d*:*

$$D(\pi_1|\pi_2) = \sum_{i=1}^{n} \pi_d(\omega_{\sigma(i)})[\Pi_1(A_{\sigma(i)}) - \Pi_1(A_{\sigma(i+1)})] \tag{5}$$

where σ *is a permutation of indices such that* $\pi_d(\omega_{\sigma(i)}) \leq ... \leq \pi_d(\omega_{\sigma(n)})$ *and* $A_{\sigma(i)} = \{\omega_{\sigma(i)}, ..., \omega_{\sigma(n)}\}$*, i=1..n and* $A_{\sigma(n+1)} = 0$*.*

Example 6. *Considering the distributions of Example 2, the application of the divergence measure gives:*
$D(\pi_1|\pi_2) = 0.49$*,* $D(\pi_1|\pi_3) = 0.3$*,* $D(\pi_1|\pi_4) = 1$*.*
Again, we obtain a different order from Example 2 and Example 4: π_3 *is the closest to* π_1 *and* π_4 *is the farthest.*

Clearly, the measure D is not symmetric. Moreover, given any possibility distribution π_i, the proposed information divergence measure gives the maximum divergence degree (Equal to 1) for all possibility distributions π_j satisfying $Inc(\pi_i \wedge \pi_j) = 1$, in other words, when the distribution difference π_d is normalized. Hence, we can no longer discriminate between these π_j's. Example 7 emphasizes this limit:

Example 7. *Let us consider the same distributions* π_1 *and* π_4 *of the previous example. Let us consider* $\pi_5[0,\ 1,\ 1,\ 1]$*.* $D(\pi_1|\pi_5) = D(\pi_1|\pi_4) = 1$*. We can conclude that this measure is not enough discriminatory since* π_4 *appears closer to* π_1 *than* π_5 *was.*

5 Information Affinity: A New Possibilistic Similarity Measure

Considering the aforementioned weaknesses related to the existing measures of divergence between possibility distributions, we will propose a new measure that

overcomes these drawbacks. The proposed measure takes into account a classical informative distance along with the well known inconsistency measure. Among the classical informative distance functions (Manhattan, Euclidean, Chebyshev, Sorensen, etc.) we choose the Manhattan distance: a simple distance which, when combined with the inconsistency measure, satisfies the expected properties mentioned in Section 3.

The choice of combining these two criteria is justified by the fact that neither the distance measure nor the inconsistency measure, taken separately, allows us to decide about the closest distribution to a given one (Example 8 emphasizes this problem). More formally, let us consider three possibility distributions π_1, π_2 and π_3. Our aim is to determine which, from π_2 and π_3, is closer to π_1. In the case of equal conflict, i.e., $Inc(\pi_1 \wedge \pi_2) = Inc(\pi_1 \wedge \pi_3)$, it is the classical distance that will decide about the closest distribution. In the same way, when we have equal distances, i.e., $d(\pi_1, \pi_2) = d(\pi_1, \pi_3)$, it is the turn of the conflict (inconsistency) measure to decide about the closest distribution, i.e., the less conflicting will be the closest.

Example 8. *Let us consider the following possibility distributions:* $\pi_1[1,\ 0,\ 0,\ 0]$, $\pi_2[0.4,\ 1,\ 0.8,\ 0.5]$, $\pi_3[0.2,\ 1,\ 1,\ 0.7]$. *If we use a classical distance measure (e.g. Manhattan distance), we obtain,* $d(\pi_1,\pi_2)=d(\pi_1,\pi_3)=\frac{2.9}{4} = 0.725$. *Hence, we can not decide wether* π_2 *or* π_3 *is closer to* π_1. *We can obtain similar situations even when using another distance (Euclidean, Chebyshev, etc.).*

Let us now consider the following possibility distributions:

$\pi'_1[1,\ 0,\ 0,\ 0]$, $\pi'_2[0,\ 1,\ 0,\ 0]$, $\pi'_3[0,\ 1,\ 1,\ 1]$.
We have $Inc(\pi'_1, \pi'_2) = Inc(\pi'_1, \pi'_3) = 1$. *Again, we can not decide which from* π'_2 *and* π'_3 *is closer to* π'_1.

Definition 4. *Let* π_1 *and* π_2 *be two possibility distributions on the same universe of discourse* Ω. *We define a measure InfoAff*(π_1, π_2) *as follows:*

$$InfoAff(\pi_1, \pi_2) = 1 - \frac{d(\pi_1, \pi_2) + Inc(\pi_1 \wedge \pi_2)}{2} \tag{6}$$

where $d(\pi_1, \pi_2) = \frac{1}{n}\sum_{i=1}^{n} |\pi_1(\omega_i) - \pi_2(\omega_i)|$ *represents the Manhattan distance between* π_1 *and* π_2 *and* $Inc(\pi_1 \wedge \pi_2)$ *tells us about the degree of conflict between the two distributions (see Equation (1)). Note that the* $\frac{1}{2}$ *value is necessary to obtain the required range [0,1].*

Two possibility distributions π_1 and π_2 are said to have a strong affinity (resp. weak affinity) if $InfoAff(\pi_1, \pi_2) = 1$ (resp. $InfoAff(\pi_1, \pi_2) = 0$).

Proposition 1. *The* $InfoAff$ *measure satisfies the six properties.*

Proofs
Property 1. Non-negativity:
By definition, $0 \leq d(a,b) \leq 1$. *Moreover,* $0 \leq Inc(a,b) \leq 1$ *(possibility degrees* $\in$ *[0,1]).* $\Rightarrow 0 \leq \frac{d(a,b)+Inc(a,b)}{2} \leq 1 \Rightarrow 0 \leq 1 - \frac{d(a,b)+Inc(a,b)}{2} \leq 1 \Rightarrow$ $InfoAff(a,b) \geq 0$.

Property 2. Symmetry:

$InfoAff(b,a) = 1 - \frac{d(b,a)+Inc(b\wedge a)}{2} = 1 - \frac{d(a,b)+Inc(a\wedge b)}{2} = InfoAff(a,b)$.

Property 3. Upper bound and Non-degeneracy:

$\forall\ b = a,\ InfoAff(a,b) = InfoAff(a,a) = 1 - \frac{d(a,a)+Inc(a\wedge a)}{2} = 1 - \frac{(0+0)}{2} = 1$.
Note that in the case of $b \neq a$, $Inc(a \wedge b)$ *could be equal to 0 but in any case we have* $d(a,b) \neq 0$, *consequently,* $InfoAff(a,b)$ *could not be equal to 1.*
Moreover, $InfoAff(a,b) = 1$ *occurs in the following two cases:*
Case 1: When $d(a,b) = 0$ *and* $Inc(a,b) = 0$, *which occurs only when* $a = b$.
Case 2: When $d(a,b) = -Inc(a,b)$ *which is impossible because* $d(a,b) \geq 0$ *and* $Inc(a,b) \geq 0$.

Property 4. Lower bound:

$InfoAff(a,b) = 0 \Leftrightarrow \frac{d(a,b)+Inc(a\wedge b)}{2} = 1 \Leftrightarrow d(a,b) + Inc(a \wedge b) = 2$.
Since $max(d(a,b)) = 1$ *and* $max(Inc(a,b)) = 1$, *then obviously we have* $d(a,b) = 1$ *and* $Inc(a,b) = 1$. *These two equalities, simultaneously hold, only when a and b are maximally contradictory, i.e, when a and b simultaneously satisfy all the following conditions: i) a and b are binary possibility distributions, ii) nor a neither b could represent total ignorance, iii) a and b should be normalized and iv) b is the negation (the complement) of a (see Property 4).*

Property 5. Inclusion:

If a is more specific than b which is in turn more specific then c, automatically, we can conclude that a, b and c are fully consistent with each others (they all share at least one state which is fully possible), i.e., Inc(a,b)=Inc(a,c)=Inc(b,c)=1. Moreover, it is obvious to see that d(a,b)≤d(a,c). So, $1 - \frac{d(a,b)+1}{2} \geq 1 - \frac{d(a,c)+1}{2}$ $\Rightarrow InfoAff(a,b) \geq InfoAff(a,c)$.

Property 6. Permutation:

Suppose that we have $InfoAff(a,b) > InfoAff(c,d)$. *Hence a', b', c' and d' are possibility distributions obtained by permuting elements having the same indexes in a, b, c and d. Since we are computing d and Inc degree by degree, the pairwise permutation of the elements has no effect on d and Inc. So we obtain d(a,b)=d(a',b') and Inc(c,d)=Inc(c',d')* $\Rightarrow InfoAff(a',b') \geq InfoAff(c',d')$.

Example 9. *Let us revisit each one of the examples listed above and see the results given by our measure for these same examples:*

Examples 2, 4 and 6: $\pi_1[1,\ 0.5,\ 0.3,\ 0.7]$, $\pi_2[1,\ 0,\ 0,\ 0]$, $\pi_3[0.9,\ 1,\ 0.3,\ 0.7]$, $\pi_4[0,\ 1,\ 0.3,\ 0.7]$.

$InfoAff(\pi_1,\pi_2) = 0.82$, $InfoAff(\pi_1,\pi_3) = 0.88$, $InfoAff(\pi_1,\pi_4) = 0.66$. *Hence,* π_3 *is the closest to* π_1 *and* π_4 *is the farthest: a different order from the ones obtained in Example 2 and 4. Note that our measure gives the same order, for this example, as the one given by the divergence measure.*

Example 3: $\pi_1[1,\ 0,\ 0,\ 0]$, $\pi_2[0,\ 1,\ 1,\ 1]$, $\pi_3[0,\ 1,\ 0,\ 1]$, $\pi_4[1,\ 0,\ 1,\ 0]$.
$InfoAff(\pi_1, \pi_2) = 0$, $InfoAff(\pi_3, \pi_4) = 0$. *$InfoAff$ is minimal for both cases: a different result from the one obtained in Example 3.*

Example 5: $\pi_1[1,\ 0,\ 0,\ 0]$, $\pi_2[1,\ 0,\ 0,\ 0]$, $\pi_3[0,\ 1,\ 1,\ 1]$.
$InfoAff(\pi_1, \pi_2) = 1$, $InfoAff(\pi_1, \pi_3) = 0$. *Hence, π_2 is the closest possible distribution to π_1 and π_3 represents the worst case. Again, we obtain a different result from the one of Example 5. Still with Example 5, if we take possibility distributions* $\pi_4[0, 1, 1, 0]$ *and* $\pi_5[0, 1, 0, 0]$, *we obtain* $InfoAff(\pi_1, \pi_4) = 0.125$ *and* $InfoAff(\pi_1, \pi_5) = 0.25$. *Hence, π_5 is closer to π_1 than π_4.*

To finish, Example 8: $\pi_1[1,\ 0,\ 0,\ 0]$, $\pi_2[0.4,\ 1,\ 0.8,\ 0.5]$, $\pi_3[0.2,\ 1,\ 1,\ 0.7]$.
$InfoAff(\pi_1, \pi_2) = 0.33$, $InfoAff(\pi_1, \pi_3) = 0.16$, $\Rightarrow$ *π_2 is closer to π_1 than π_3.*
If we take: $\pi'_1[1,\ 0,\ 0,\ 0]$, $\pi'_2[0,\ 1,\ 0,\ 0]$, $\pi'_3[0,\ 1,\ 1,\ 1]$.
$InfoAff(\pi'_1, \pi'_2) = 0.25$, $InfoAff(\pi'_1, \pi'_3) = 0$, $\Rightarrow$ *π'_2 is closer to π'_1 than π'_3.*

6 Practical Applications of Information Affinity

We mention some fields in which Information Affinity measure could be useful.

6.1 Machine Learning: Classification and Clustering

The proposed information affinity measure could be used in many classification and clustering algorithms, especially in those using possibility theory as a tool for dealing with existing uncertainty in the learning process [12] [13]. For instance, $InfoAff$ could be used as the basis of an attribute selection measure for inducing decision trees from imprecisely labeled data. More formally, it will allow to select the attribute that, when chosen, will provide partitions of the training set containing maximally similar instances, i.e, instances having as much as possible similar possibility distributions on their classes. Still in classification problems, $InfoAff$ could be also used in most of distance based classifiers which are induced from imprecise data, e.g. k-nearest neighbor classifiers, genetic algorithms, artificial immune recognition systems, etc. Likewise, $InfoAff$ could be used in possibilistic clustering [15] as the distance criterion which will allow to decide about the belonging or not of an instance to a given cluster which is characterized by a possibility distribution.

6.2 Evaluation of Possibilistic Classifiers

The use of our measure does not only comply with learning, it could also be used in the evaluation of possibilistic classifiers. Recall that within a possibilistic classifier, the classification result is given in the form of a possibility distribution (π^{res}) on the different possible classes of the problem ($\Omega = \{C_1, C_2, ..., C_n\}$).

Generally, the well known percentage of correct classification (PCC) is used to evaluate classifiers ($PCC = \frac{nbr\_well\_classified\_inst}{total\_nbr\_classified\_inst} \times 100$). In the possibilistic setting, it is used as follows: it chooses for each classified instance the class having

the highest possibility degree (equal to 1). If more than one class is obtained, then one of them is chosen randomly. The obtained class is considered as the class of the testing instance. Consequently, $nbr\_well\_classified\_inst$ corresponds to the number of testing instances for which the class obtained by the possibilistic classifier (the more plausible class) is the same as the real class.The limitation of this adaptation of the PCC criterion to the possibilistic setting, is that it chooses randomly one of the more plausible classes which may miss-classify some instances. Moreover, even when there is only one more plausible class, focusing on that class and ignoring the rest of the classes (classes with possibility degrees different from 1) is problematic. In fact, ignoring the rest of the degrees implies ignoring a part of the information given by the resulting possibility distribution (π^{res}).

So, a solution is to define an affinity based criterion PCC_Aff (Equation (7)) which takes into account the mean affinity relative to all the classified testing instances: the average of the similarities between the resulting possibility distribution (π^{res}) and the real (completely sure) possibility distribution (π^j) of each classified instance $I_j,\ j = 1..n$. When PCC_Aff is close to 100%, the classifier is good whereas when it falls to 0%, it is considered as a bad classifier.

$$PCC\_Aff = \frac{\sum_{j=1}^{n} InfoAff(\pi^{res}, \pi^j)}{total\_nbr\_classified\_inst} \times 100 \tag{7}$$

Note that an alternative PCC criterion for possibilistic classifiers, more precisely, for possibilistic decision trees was proposed in [2]. The so-called *Qualitative PCC* denoted by Q_PCC is different from PCC_Aff: the former is based on an Euclidean distance between the real (completely sure) possibility distribution of each classified instance and its resulting qualitative possibility distribution which is induced from the leximin-leximax ordering on the different classes given by the tree.

6.3 Comparing Opinions and Sensor Measures

In many situations, comparing opinions of different agents supports decision making. For instance, suppose we have a group of candidates taking part in a competitive entry examination. Each candidate will be asked questions. Some flexibility is offered to the candidates which will allow them to give a possibility degree for each proposed response instead of giving a precise response. The final best candidate will be the one giving possibility distributions which are the most similar to the true responses (possibility distributions corresponding to completely sure knowledge).

Another interesting use of the Information Affinity measure appears for sensor diagnosis. Suppose that we have many sensors measuring a given variable. These sensors are allowed to give measures with some errors, consequently, one can represent their outputs as possibility distributions over the different possible values of the variable under study. Suppose that we are sure that a given sensor s_0 is reliable (a new installed sensor). One should compare measures (the possibility

distributions) given by the different sensors with the one given by s_0 and reject or replace those giving different measures to a certain extent.

7 Conclusion

This paper focuses on measuring the similarity between possibilistic uncertain information. One should note that, contrary to what has been done in other uncertainty formalisms, few works have been done in this direction for the case of possibility theory. After proposing some natural properties of a similarity measure between possibility distributions, after studying some few existing measures and showing their limits by examples, we have proposed a new similarity measure which takes its roots from both the measure of inconsistency and a classical distance. We have contrasted our measure with the existing ones and have shown that it represents a reliable measure which recovers the limits of the few existing ones. Potential applications of the proposed measure have been mentioned in the end of the paper.

References

1. Abellan, J., Gomez, M.: Measures of divergence on credal sets. Fuzzy Sets and Systems 157, 1514–1531 (2006)
2. Ben Amor, N., Benferhat, S., Elouedi, Z.: Qualitative classification and evaluation in possibilistic decision trees. In: FUZZ-IEEE'04 (2004)
3. Bouchon-Meunier, B., Rifqi, M., Bothorel, S.: Towards general measures of comparison of objects. Fuzzy sets and systems 84(2), 143–153 (1996)
4. Chan, H., Darwiche, A.: A distance measure for bounding probabilistic belief change. International Journal of Approximate Reasoning 38, 149–174 (2005)
5. Choquet, G.: Theory of capacities. Annales de L'Institut Fourier 54, 131–295 (1953)
6. De Baets, B., De Meyer, H.: Transitivity-preserving fuzzification schemes for cardinality-based similarity measures. EJOR 160(1), 726–740 (2005)
7. Dubois, D., Prade, H.: Possibility theory: An approach to computerized processing of uncertainty. Plenum Press, New York (1988)
8. Fixsen, D., Mahler, R.P.S.: The modified Dempster-Shafer approach to classification. IEEE. Trans. Syst. Man and Cybern. 27, 96–104 (1997)
9. Fono, L.A., Gwet, H., Bouchon-Meunier, B.: Fuzzy implication operators for difference operations for fuzzy sets and cardinality-based measures of comparison. EJOR 183, 314–326 (2007)
10. Higashi, M., Klir, G.J.: Measures of uncertainty and information based on possibility distributions. Int. J. General Systems 9(1), 43–58 (1983)
11. Higashi, M., Klir, G.J.: On the notion of distance representing information closeness: Possibility and probability distributions. IJGS 9, 103–115 (1983)
12. Hüllermeier, E.: Possibilistic instance-based learning. AI 148(1-2), 335–383 (2003)
13. Jenhani, I., Ben Amor, N., Elouedi, Z., Mellouli, K.: Decision Trees as Possibilistic Classifiers (paper submitted)
14. Jousselme, A.L., Grenier, D., Bossé, E.: A new distance between two bodies of evidence. Information Fusion 2, 91–101 (2001)

15. Krishnapuram, R., Keller, J.M.: A possibilistic approach to clustering. IEEE Transactions on Fuzzy Systems 1(2), 98–110 (1993)
16. Kroupa, T.: Measure of divergence of possibility measures. In: Proceedings of the 6th Workshop on Uncertainty Processing, Prague, pp. 173–181 (2003)
17. Kullback, S., Leibler, R.A.: On information and sufficiency. Annals of Mathematical Statistics 22, 79–86 (1951)
18. Sanguesa, R., Cabos, J., Cortes, U.: Possibilistic conditional independence: a similarity based measure and its application to causal network learning. IJAR (1997)
19. Sanguesa, R., Cortes, U.: Prior knowledge for learning networks in non-probabilistic settings. IJAR 24, 103–120 (2000)
20. Shafer, G.: A mathematical theory of evidence. Princeton University Press, Princeton (1976)
21. Tessem, B.: Approximations for efficient computation in the theory of evidence. Artificial Intelligence 61, 315–329 (1993)
22. Tversky, A.: Features of similarity. Psychological Review 84, 327–352 (1977)
23. Wang, X., De Baets, B., Kerre, E.: A comparative study of similarity measures. Fuzzy Sets and Systems 73(2), 259–268 (1995)
24. Williams, M-A.: An Operational Measure of Similarity for Belief Revision Systems (1997)
25. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets ans Systems 1, 3–28 (1978)
26. Zouhal, L.M., Denoeux, T.: An evidence-theoric k-NN rule with paprameter optimization. IEEE Trans. Syst. Man Cybern. C 28(2), 263–271 (1998)

Measuring the Quality of Health-Care Services: A Likelihood-Based Fuzzy Modeling Approach

Giulianella Coletti[1], Luca Paulon[2],
Romano Scozzafava[3], and Barbara Vantaggi[3]

[1] Università di Perugia (Italy)
coletti@dipmat.unipg.it
[2] Associazione Fatebenefratelli per la Ricerca, Roma (Italy)
luca.paulon@afar.it
[3] Università di Roma "La Sapienza" (Italy)
{romscozz,vantaggi}@dmmm.uniroma1.it

Abstract. We face the problem of constructing a model which is suited for an effective evaluation of the quality of a health–care provider: to this purpose, we focus on some relevant indicators characterizing the various services run by the provider. We rely on a fuzzy modeling approach by using the interpretation (in terms of coherent conditional probability) of a membership function of a fuzzy set as a suitable likelihood.

Keywords: Health-care services, efficiency, fuzzy modeling.

1 Introduction

Our aim is to realize, by focusing on some relevant aspects, a model which is suited for an effective evaluation of the quality of a health–care provider, e.g. clinical department, day hospital, etc..

We refer to the term "efficient", that can be looked on as a fuzzy concept. If the health–care provider runs several services (e.g.: supply of clinical tests, delivering of medical therapies, etc.), the efficiency of every service can be defined and characterized by a certain number k of key "indicators" (e.g.: waiting time necessary to get the required clinical test, cost of treatment per outpatient episode, etc.): denote by X_j the j-th indicator, which is then a component of the vector $X = (X_1, X_2, \ldots, X_k)$.

Using the interpretation of membership of a fuzzy set as a suitable likelihood (the main points of this approach are reported in Sections 2.1 and 2.2), we can ask all the medical staff (e.g., the doctors) to claim their (subjective) judgment on a given service. Considering the event $E =$ "a doctor claims efficient the service" and x ranging over a subset of the cartesian product of the ranges of $X_1, X_2, \ldots, X_k$ (*i.e.* over all possible realizations of the vector X), we assign the membership function $\mu_E(x)$ by taking it equal to the probability of the conditional event $E|\{X = x\}$. This probability is assessed through the following procedure: the doctors of the medical staff are required to evaluate the degree

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 853–864, 2007.

of efficiency of the service (given $X = x$) by a number between 0 and 1, so we get a vector $(d_1, d_2, \ldots, d_n)$ (where n is the number of doctors); then, putting

$$d = \frac{d_1 + d_2 + \ldots + d_n}{n} \ ,$$

we assess $P(E|\{X = x\}) = d$.

The same procedure can be applied to patients to find the analogous membership function $\mu_F(x)$ as a conditional probability, by considering, now, the event $F =$ "a patient claims efficient the service".

An important setup of the model can be reached by relying on the concept of *similarity* (recalled in Section 2.3): once the two membership functions $\mu_E(x)$ and $\mu_F(x)$ are assessed, we may check whether they are "similar" more than a given threshold, and only in this case decide to proceed with their "aggregation"; otherwise it may be interesting to go back to the interviews to doctors and patients to try to explain the reasons for this gap (or, alternatively, to shed light on how to choose a suitable procedure of aggregation).

A convex combination $\mu_G(x)$ of the two membership functions $\mu_E(x)$ and $\mu_F(x)$ turns out to be a coherent extension – see Section 2.2 – of the above conditional probability $P(\cdot|\cdot)$ assessed on the set

$$\left\{E|\{X = x\}\,, F|\{X = x\}\right\}$$

to a "new" conditional event $G|\{X = x\}$, with G such that $E \wedge F \subseteq G \subseteq E \vee F$. We can then obtain a fuzzy model relative to all services of the provider as a weighted mean of the memberships $\mu_G(x)$ measuring the efficiency of all different services.

Finally, denoting by $f_x^{(s)}$ the relative frequency, for the patients attending the service s, of the realization x, and by $\mu_G^{(s)}(x)$ the relevant membership, we can measure the efficiency of the provider by means of the "expected value" (balanced score)

$$h = \sum_s g_s \sum_x f_x^{(s)} \mu_G^{(s)}(x)\,,$$

where each g_s is a suitable "weight" relative to the service s in the health–care provider.

In the last Section we give preliminary results concerning an exploratory case study: we thank dr. Alessandra Campolongo and dr. Alessandro Angeli for their valuable help in the collection of data.

2 Previous Results

2.1 Fuzzy Sets and Conditional Probability

We refer to the state of information (at a given moment) of a real (or fictitious) person that will be denoted by "You". If X is a (not necessarily numerical) quantity with range C_X, let A_x be, for any $x \in C_X$, the event $\{X = x\}$. The family $\{A_x\}_{x \in C_x}$ is obviously a *partition* of the certain event $\Omega = C_X$.

Now, let φ_X be any *property* related to the quantity X: from a pragmatic point of view, it is natural to think that You have some information about possible values of X, which allows You to refer to a suitable *membership function* of the fuzzy subset of "elements of C_X with the property φ_X".

For example, if X is a numerical quantity and φ_X is the property "small", for You the membership function $\mu_\varphi(x)$ may be put equal to 1 for values x of X less than a given x_1, while it is put equal to 0 for values greater than x_2; then it is taken as decreasing from 1 to 0 in the interval from x_1 to x_2: this choice of the membership function implies that, for You, elements of C_X less than x_1 have the property φ_X, while those greater than x_2 do not.

So the real problem is that You are doubtful (and so uncertain) on having or not the property φ_X those elements of C_X between x_1 and x_2. Then the interest is in fact directed toward *conditional events* such as $E|A_x$, where x ranges over the interval from x_1 to x_2, with

$$E=\{\text{You claim that } X \text{ has the property } \varphi_X\}\,, \quad A_x=\{\text{the value of } X \text{ is } x\}.$$

It follows that, while You may assign to each of these conditional events a degree of belief (subjective probability) $P(E|A_x)$, You must not assign a degree of belief $1-P(E|A_x)$ to the event E under the assumption A_x^c (the value of X is not x), since an additivity rule *with respect to the conditioning events* does not hold. In other words, it seems sensible to identify the values of the membership function $\mu_\varphi(x)$ with suitable conditional probabilities. In particular, putting

$$H_o = \{X \text{ is greater than } x_2\}\,, \qquad H_1 = \{X \text{ is less than } x_1\},$$

one has that E and H_o are incompatible and that H_1 implies E, so that, by the rules of a conditional probability, $P(E|H_o)=0$ and $P(E|H_1)=1$.

Notice that this conditional probability $P(E|A_x)$ is *directly* introduced as a function on the set of conditional events (and without assuming any given algebraic structure). Is that possible? In the usual (Kolmorogovian) approach to conditional probability the answer is NO, since the introduction of $P(E|A_x)$ would require the consideration (and the assessment) of $P(E \wedge A_x)$ and $P(A_x)$ (assuming positivity of the latter). But this *could not be* a simple task: in fact in this context the only sensible procedure is to assign directly $P(E|A_x)$. For example, it is possible to assign the (conditional) probability that "You claim this number is small" knowing its value x, but not necessarily the probability that "The number has the value x" (not to mention that, for many choices of the quantity X, the corresponding probability can be equal to zero). These problems are easily by–passed in our framework.

2.2 Coherent Conditional Probability

Our approach to probability (expounded, e.g., in [3] and [4]: see also the book [6]) is based on *coherence* (a concept that goes back to de Finetti [11]). The starting point is a synthesis of the available information, expressed by one or more *events*: to this purpose, the concept of event must be given its more general meaning, *i.e. it must not be looked on just as a possible outcome* (a subset of the so–called "sample space"), but expressed by a *proposition*. Moreover, events

play a two–fold role, since we must consider not only those events which are the direct object of study, but also those which represent the relevant "state of information": in fact a "bunch" of *conditional* events, together with a relevant "partial" assessment of *conditional* probability, are the objects that allow to manage specific (conditional) situations and to update degrees of belief on the basis of the evidence.

The role of coherence is in fact that of ruling an extension process, starting from the classic *axioms for a conditional probability*. Given a set $\mathcal{C} = \mathcal{G} \times \mathcal{B}^o$ of conditional events $E|H$ such that $\mathcal{G}$ is a Boolean algebra and $\mathcal{B} \subseteq \mathcal{G}$ is closed with respect to (finite) logical sums, and putting $\mathcal{B}^o = \mathcal{B} \setminus \{\emptyset\}$, then $P : \mathcal{C} \to [0, 1]$ is such that

(i) $P(H|H) = 1$, for every $H \in \mathcal{B}^o$,

(ii) $P(\cdot|H)$ is a (finitely additive) probability on $\mathcal{G}$ for any given $H \in \mathcal{B}^o$,

(iii) $P\big((E \wedge A)|H\big) = P(E|H) \cdot P\big(A|(E \wedge H)\big)$, for every E, $A \in \mathcal{G}$ and E, $E \wedge H \in \mathcal{B}^o$.

A peculiarity (which entails a large flexibility in the management of any kind of uncertainty) of this approach to conditional probability is that, due to its *direct* assignment as a whole, the knowledge (or the assessment) of the "joint" and "marginal" unconditional probabilities $P(E \wedge H)$ and $P(H)$ is not required; moreover, the *conditioning* event H (which *must* be a *possible* event) may have *zero probability*.

A conditional probability P is defined on $\mathcal{G} \times \mathcal{B}^o$: however it is possible, through the concept of *coherence*, to handle also those situations where we need to assess P on an *arbitrary* set of conditional events $\mathcal{C} = \{E_1|H_1, \ldots, E_n|H_n\}$.

Definition. The assessment $P(\cdot|\cdot)$ on $\mathcal{C}$ is *coherent* if there exists $\mathcal{C}' \supset \mathcal{C}$, with $\mathcal{C}' = \mathcal{G} \times \mathcal{B}^o$ ($\mathcal{G}$ Boolean algebra and $\mathcal{B} \subseteq \mathcal{G}$ closed with respect to logical sums) such that P can be extended from $\mathcal{C}$ to $\mathcal{C}'$ as a *conditional probability*.

Concerning coherence, another fundamental result is the following.

Theorem. *Let $\mathcal{K}$ be any family of conditional events, and take an* arbitrary *family $\mathcal{C} \subseteq \mathcal{K}$. Let P be an assessment on $\mathcal{C}$; then there exists a (possibly not unique) coherent extension of P to $\mathcal{K}$* if and only if *P is coherent on $\mathcal{C}$.*

Coherence of conditional assessments can be ruled by a fundamental *characterization theorem*, which is based on checking the compatibility of a suitable sequence of linear systems: for the sake of brevity, we avoid here any detail or deepening, and we just mention that a coherent conditional probability on an arbitrary family $\mathcal{C}$ can be characterized by suitably representing it (in any finite subset $\mathcal{F}$ of $\mathcal{C}$) by means of a *class* $\{P_\alpha\}$ of coherent *unconditional* probabilities giving rise to the so-called *zero-layers* (indexed by α). See, e.g., the book [6], and for a concrete application – to medical diagnosis – see [5].

Finally, notice that what is usually emphasized in the relevant literature – when a conditional probability $P(E|H)$ is taken into account – is only the fact that $P(\cdot|H)$ *is a probability for any given* H: this is a very restrictive (and mis-

leading) view of conditional probability, corresponding trivially to just a modification of the so-called "sample space" Ω. It is instead essential to regard also the conditioning event H as a "variable", *i.e.* the "status" of H in $E|H$ is not just that of something representing a given *fact*, but that of an (uncertain) *event* (like E) for which the knowledge of its truth value is not required. In other words, even if beliefs may come from various sources, they can be treated in the same way, since the relevant *conditioning* events (including both *statistical data* and also – to use Zadeh's [18] terminology – *perception–based information*) can always be considered as being *assumed* propositions.

Roughly speaking, looking on a coherent conditional probability – acting as a membership function – as a general non-additive "uncertainty" measure $m(\cdot) = P(E|\cdot)$ of the conditioning events amounts to referring to what in the statistical jargon is called "likelihood", with its various "ad hoc" extensions from a point function to a set function. The problem is that without a clear, precise and rigorous mathematical frame, the likelihood "per se" is not a proper tool to deal with statistical inference and to manage partial and vague information (these aspects are discussed more deeply in [8]).

2.3 Similarity

For a deepening of our approach to fuzzy sets and for other relevant formal definitions, see [7], where we show not only how to define fuzzy subsets, but we also introduce in a very natural way the counterparts of the basic continuous T-norms and the corresponding dual T-conorms, bound to the former by *coherence*.

In the sequel, a fuzzy subset of C_X (relative to the property φ) will be identified with its membership function $\mu_\varphi(x) = P(E|A_x)$ and denoted by E^*_φ.

We recall now the following definition of *similarity*, taken from [2], where the term "resemblance" is used. For more details, see [17].

Let $\mathcal{F}(C_X)$ be the family of fuzzy subsets E^*_π of C_X. A *similarity* S is a mapping

$$S : \mathcal{F}(C_X) \times \mathcal{F}(C_X) \longrightarrow [0, 1]$$

such that

1. (Symmetry) $S(E^*_\varphi, E^*_\psi) = S(E^*_\psi, E^*_\varphi)$;
2. (Reflexivity) $S(E^*_\varphi, E^*_\varphi) = 1$.

Proposition ([17])**.** Given any two fuzzy subsets E^*_φ and E^*_ψ of $\mathcal{C}_X$, with

$$\mu_\varphi(\cdot) = P(E_\varphi|\cdot) \ , \ \mu_\psi(\cdot) = P(E_\psi|\cdot)\,,$$

let $P(E_\varphi \wedge E_\psi|A_x)$ be a relevant coherent assessment. Then any coherent extension of $P(\cdot|\cdot)$ to the conditional event $(E_\varphi \wedge E_\psi)|(E_\varphi \vee E_\psi)$ is a similarity.

The existence of such a function is warranted by the fundamental extension Theorem recalled in Section 2.2. The semantic behind this choice is the following: **the more** two fuzzy subsets are considered to be similar, **the more** if You claim *at least one* of the two corresponding properties You are willing to claim *both* properties.

How to compute $S(E^*_\varphi, E^*_\psi)$? Given $\mu_\varphi(\cdot) = P(E_\varphi|\cdot)$ and $\mu_\psi(\cdot) = P(E_\psi|\cdot)$, the membership functions $\mu_{\varphi\cup\psi}(\cdot)$ and $\mu_{\varphi\cap\psi}(\cdot)$ of the fuzzy sets $(E^*_\psi \cup E^*_\varphi)$ and $(E^*_\psi \cap E^*_\varphi)$ (corresponding to a T–conorm and a dual T–norm, see [7]) arise as coherent extensions of the assessment P given on $\{E_\psi|A_x\,,\ E_\varphi|A_x\ :\ A_x \in C_X\}$.

Then, given a conditional probability $P(\cdot|\cdot)$ on $\mathcal{A}_X \times \mathcal{A}^o_X$ (which gives rise to a class $\{P_\alpha\}$ of coherent unconditional probabilities, see Section 2.2), we have (for simplicity we refer to a finite $\mathcal{C}_X$)

$$S(E^*_\varphi, E^*_\psi) = \frac{\sum_x \mu_{\varphi\cap\psi}(x)\lambda_\alpha(x)}{\sum_x \mu_{\varphi\cup\psi}(x)\lambda_\alpha(x)} \tag{1}$$

where $\lambda_\alpha(x) = P_\alpha(A_x)$, with α the zero-layer of the event $E_\psi \vee E_\varphi$.

Notice that (contrary to what happens in the classic fuzzy framework) this approach to similarity is able to take into account – through the probability values $\lambda_\alpha(x)$ – possible different "weights" of the values x.

In [17] it is shown how some classic similarity functions (the most used in applications and proposed in the relevant literature) are related to the above formula involving conditional probability.

Given a T–norm, the T–transitivity property for a similarity S reads as follows:

$$S(E^*_\varphi, E^*_\psi) \geq T(S(E^*_\varphi, E^*_\nu), S(E^*_\nu, E^*_\psi)),$$

for any $E^*_\varphi, E^*_\psi, E^*_\nu \in \mathcal{L}(\mathcal{C}_X)$. The similarity S does not generally satisfy the T-transitivity property for some suitable T-norm, for example for $T = min$.

However, the similarity S given by (1) satisfies the T-transitivity property for T=T_L (Łukasiewicz T-norm), i.e.

$$S(E^*_\varphi, E^*_\psi) \geq \max\{0\,,\ S(E^*_\varphi, E^*_\nu) + S(E^*_\nu, E^*_\psi) - 1\}$$

for any E^*_φ,E^*_ψ,$E^*_\nu \in \mathcal{L}(C_X)$, as proved in [17].

Concerning transitivity a relevant reference is [9].

3 Measuring the Quality of Health–Care Services

Judgments about the present or future quality of health-care services are usually made on the basis of observations about the performance of providers. As far as we know, in the relevant literature there seems to be lack of consensus about the way of measuring the quality of services, and how such an important notion should be meaningfully formalized. We cite here only a few papers, such as [1], [12], [13], [14], [15], [16].

We start by pointing out that we refer to an health–care organization that has a governing body, an organized medical staff, a professional staff and inpatient facilities, and provides medical, nursing, and related services for ill and injured patients 24 hours per day, seven days per week.

Let us consider a vector $X = (X_1, X_2, \ldots, X_k)$ relative to the performance of a specific service of a health–care provider. For example, referring to the

treatment of stroke in a neuroscience department and using the notation introduced in Section 1, as key indicators (with respect to a given period) of the efficiency of this department we may single–out, e.g., $X_1 =$"number of patients rehabilitated", $X_2 =$"number of patients treated with platelet inhibitor", $X_3 =$"average duration of stay", $X_4 =$"number of patients treated with anticoagulants", $X_5 =$"number of patients dead within 30 days",..., and so on.

We denote by x a possible realization of the vector X. Clearly, x belongs to a subset $\mathcal{C}$ of the cartesian product of the ranges of $X_1, X_2, \ldots, X_k$.

To construct a membership function relative to the *efficiency* of a given service we may, for each possible – observed or assumed – x, ask the members of the medical staff how much they are willing to claim the service efficient, given the value x. Considering the event $E =$ "a doctor claims efficient the service", the membership function $\mu_E(x)$ is assigned by taking it equal to the probability of the conditional event $E|\{X = x\}$, and this probability is assessed through the procedure explained in the Introduction.

The same procedure can be applied to the patients to find the analogous membership function $\mu_F(x)$, so that

$$\mu_E(x) = P\big(E|\{X = x\}\big) \ , \ \ x \in \mathcal{C} \tag{2}$$

$$\mu_F(x) = P\big(F|\{X = x\}\big) \ , \ \ x \in \mathcal{C} \ . \tag{3}$$

Once the two membership functions $\mu_E(x)$ and $\mu_F(x)$ are assessed, we proceed with a suitable "aggregation" to measure the overall (with respect to both doctors and patients) efficiency of the service.

What do we mean by "suitable"? Of course, it involves the policy of the hospital (for some services it can be more important the point of view of doctors than that of patients, for others the situation can be opposite).

In other words, we should consider an event G such that

$$E \wedge F \subseteq G \subseteq E \vee F.$$

It could be interpreted as "something in between" with respect to either a complete agreement in the judgments of (the "average") doctor and patient (this situation corresponds to the intersection $E \wedge F$), or a situation (corresponding to the union $E \vee F$) in which at least one of them (possibly both) claims the efficiency of the relevant service. Our interpretation of this event should correspond to a "fair referee" (or expert) that, being aware of what doctor and patient claim, has a "calibrated" opinion. In particular, when both doctor and patient claim the efficiency of the service, the expert does so as well, while if he claims its efficiency, it means that either doctor or patient (possibly both) agree with this claim.

So the real problem is to assess coherently the probability (conditional to $\{X = x\}$) of the event G representing the aggregation, i.e. a syntactic one. Put

$$P(E|\{X = x\}) = e \ , \ \ P(F|\{X = x\}) = f \, ; \tag{4}$$

then it can be easily proved that coherent assessments of the probabilities $P(V)$ and $P(U)$, with $V = (E \wedge F)|\{X = x\}$ and $U = (E \vee F)|\{X = x\}$, must satisfy the following inequalities

$$\max\{e + f - 1, 0\} \leq P(V) \leq \min\{e, f\}\,,$$

$$\max\{e, f\} \leq P(U) \leq \min\{e + f, 1\}\,.$$

Then, coherence entails

$$\max\{e + f - 1, 0\} \leq P(G|\{X = x\}) \leq \min\{e + f, 1\}\,.$$

For simplicity and taking into account the above discussion we can assess

$$P(G|\{X = x\}) = \gamma\mu_E(x) + (1 - \gamma)\mu_F(x) \tag{5}$$

with $\gamma \in [0, 1]$. This amounts to introducing the "aggregated" membership function $\mu_G(x) = P(G|\{X = x\})$ as a convex combination. By solving a relevant linear system (as recalled in Section 2), it can be proved that the assessment (5) is coherent, *i.e.* it is a coherent extension of the two initial assessments – relative to doctors and patients – of the conditional probabilities $P(E|\{X = x\})$ and $P(F|\{X = x\})$.

Denoting by f_x the relative frequency of the realization x, we can measure the efficiency $\mathcal{E}_s$ of the service s by the formula

$$\mathcal{E}_s = \sum_x f_x \mu_G(x) = \sum_x P(X = x) P(G|\{X = x\})\,; \tag{6}$$

notice that it represents the probability of the "fuzzy event" G.

Finally, the efficiency of the health–care provider can be computed by means of the balanced score, as explained in the final part of the Introduction. Here again, as in the case of choosing the weights (for each given service) relative to doctors or patients, the choice of each g_s relative to the different services depends on the specific situations concerning the given provider.

4 An Exploratory Case Study: Preliminary Results

Our preliminary study refers to an exploratory design involving both "normative" and "descriptive" aspects. First of all, we realized that doctors and patients were quite reluctant to give a numerical estimate of the *degree* of efficiency (of a given service, and assuming a value x of the key indicators' vector). It is more natural for them to face direct questions requiring "crisp" answers about both efficiency (that may depend, for example, on the importance given to a component of the vector x) and inefficiency (giving more importance, possibly, to another component of x). In fact, the second question concerning inefficiency has been posed to "measure" (in a sense) the reliability of the first answer concerning efficiency. So the *descriptive* aspects (the real behavior of interviewed

persons in front of a questionnaire) do not seem to conflict with the *normative* rule that a fuzzy concept and its negation are not complementary.

Then, for a given service and for each interviewed person we have four possible answer concerning the pair

(efficient, inefficient),

that is: (YES, NO), (YES, YES), (NO, NO), (NO, YES). We interpret the first and the last answer as representing, respectively, the maximum degree (equal to 1) or the minimum degree (equal to 0) of efficiency. On the other hand, for the two pairs (YES, YES) and (NO, NO) we can choose either to assign a degree 1/2 to both answers, or (giving different "weights" to positive or negative answer) to assign a degree 3/4 to (YES, YES) and a degree 1/4 to (NO, NO).

Our data have been collected in the department of neurology of the hospital "Fatebenefratelli" in Rome. The results of the evaluations of doctors' and patients' memberships $\mu_E(x)$ and $\mu_F(x)$, with reference to four services denoted by A, B – two kinds of ENG (electroneurography) – and C, D – two kinds of EMG (electromyography) – are reported in Table 1 and 2, respectively for the choice interpreted as

Table 1. Case (i)

	Doctor's Membership $\mu_E(x)$				Patient's Membership $\mu_F(x)$			
$A_x = t_w \times t_s$	A	B	C	D	A	B	C	D
$[0, 15] \times [0, 10]$	0.9286	0.5714	0.8750	0.7500	0.6842	0.8611	0.7778	0.8684
$[15, 30] \times [0, 10]$	0.7857	0.7143	0.8125	1.0000	0.6579	0.6944	0.6667	0.4737
$[30, 60] \times [0, 10]$	0.7143	0.6429	0.5625	0.6667	0.2368	0.2500	0.1667	0.3421
$[60, 120] \times [0, 10]$	0.2143	0.0714	0.4375	0.3333	0.2632	0.3333	0.2778	0.3684
$[> 120] \times [0, 10]$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$[0, 15] \times [10, 15]$	0.7857	0.5714	0.7500	0.6667	0.8684	0.8333	0.8333	0.9474
$[15, 30] \times [10, 15]$	0.7857	0.8571	0.7500	0.9167	0.7632	0.8333	0.7500	0.8158
$[30, 60] \times [10, 15]$	0.3571	0.2857	0.3125	0.4167	0.6579	0.6389	0.6111	0.5263
$[60, 120] \times [10, 15]$	0.2143	0.2143	0.3125	0.4167	0.6579	0.5833	0.5000	0.5526
$[> 120] \times [10, 15]$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$[0, 15] \times [15, 20]$	0.5000	0.4286	0.6875	0.6667	0.8158	0.8611	0.7778	0.9211
$[15, 30] \times [15, 20]$	0.5714	0.6429	0.8125	0.8333	0.8158	0.8333	0.8824	0.9474
$[30, 60] \times [15, 20]$	0.3571	0.4286	0.5000	0.4167	0.4474	0.4444	0.4167	0.3947
$[60, 120] \times [15, 20]$	0.0000	0.0000	0.1250	0.1667	0.2895	0.3889	0.3611	0.3158
$[> 120] \times [15, 20]$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$[0, 15] \times [20, 30]$	0.5000	0.4286	0.6875	0.6667	0.7368	0.8889	0.7500	0.7632
$[15, 30] \times [20, 30]$	0.4286	0.5000	0.8125	0.6667	0.5000	0.8611	0.6111	0.8421
$[30, 60] \times [20, 30]$	0.3571	0.3571	0.4375	0.3333	0.3421	0.5833	0.4722	0.3947
$[60, 120] \times [20, 30]$	0.1429	0.1429	0.3750	0.2500	0.2632	0.2778	0.3056	0.1842
$[> 120] \times [20, 30]$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$[0, 15] \times [> 30]$	0.3571	0.4286	0.4375	0.5000	0.8158	0.6667	0.5556	0.7368
$[15, 30] \times [> 30]$	0.3571	0.4286	0.3125	0.5000	0.6579	0.6111	0.5000	0.6053
$[30, 60] \times [> 30]$	0.2143	0.2857	0.5000	0.4167	0.3421	0.3333	0.3333	0.2632
$[60, 120] \times [> 30]$	0.0000	0.2857	0.0625	0.0000	0.2105	0.1667	0.2778	0.3158
$[> 120] \times [> 30]$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

(i) "degree 1/2 for both (YES, YES) and (NO, NO)"

or the choice interpreted as

(ii) "degree 3/4 for (YES, YES) and degree 1/4 for (NO, NO)".

As key indicators we have taken t_w (waiting time before the coverage of the service) and t_s (duration of the service). The corresponding notation (first column in each table) to single–out the pair (t_w, t_s) is of the kind $[a, b] \times [c, d]$.

In Table 3 we report the empirical probability distribution (which is not necessarily uniform, as in the classic fuzzy approach) of the events A_x (two–dimensional vectors whose components are t_w and t_s). This distribution is obtained as the observed frequency of the aforementioned pairs.

The values of the probabilities of the A_x's are necessary to compute *similarity* between $\mu_E(x)$ and $\mu_F(x)$, according to formula (1), Section 2.3: the properties φ and ψ correspond to the efficiency claimed, respectively, by doctors and patients, and as norm and conorm we choose, respectively, the classic min and max; notice that zero–layers here are not involved, since the event $E_\psi \vee E_\varphi$ has positive

Table 2. Case (ii)

	Doctor's Membership $\mu_E(x)$				Patient's Membership $\mu_F(x)$			
$A_x = t_w \times t_s$	A	B	C	D	A	B	C	D
$[0, 15] \times [0, 10]$	0.8929	0.5714	0.8750	0.7083	0.7632	0.8472	0.8333	0.8816
$[15, 30] \times [0, 10]$	0.8214	0.7857	0.7813	1.0000	0.5921	0.6528	0.5833	0.3947
$[30, 60] \times [0, 10]$	0.6429	0.6071	0.5938	0.7500	0.2500	0.2083	0.1389	0.2763
$[60, 120] \times [0, 10]$	0.1786	0.1071	0.4688	0.3333	0.2368	0.2500	0.1944	0.3158
$[> 120] \times [0, 10]$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$[0, 15] \times [10, 15]$	0.7500	0.5714	0.7500	0.6667	0.8816	0.8333	0.8889	0.9737
$[15, 30] \times [10, 15]$	0.8214	0.9286	0.8125	0.9583	0.7763	0.8056	0.7639	0.8026
$[30, 60] \times [10, 15]$	0.4643	0.3571	0.3438	0.5417	0.6184	0.5417	0.5833	0.4211
$[60, 120] \times [10, 15]$	0.3214	0.2500	0.4063	0.5417	0.6974	0.5694	0.5278	0.5395
$[> 120] \times [10, 15]$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$[0, 15] \times [15, 20]$	0.6071	0.5714	0.7813	0.6667	0.7763	0.8194	0.8056	0.9079
$[15, 30] \times [15, 20]$	0.7143	0.7500	0.8438	0.9167	0.8289	0.8611	0.9412	0.9474
$[30, 60] \times [15, 20]$	0.5357	0.5000	0.5625	0.5417	0.3816	0.3889	0.3472	0.3026
$[60, 120] \times [15, 20]$	0.0000	0.0000	0.1250	0.1667	0.2500	0.3611	0.2917	0.2632
$[> 120] \times [15, 20]$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$[0, 15] \times [20, 30]$	0.6071	0.5714	0.7813	0.6667	0.7368	0.8611	0.7639	0.7237
$[15, 30] \times [20, 30]$	0.5000	0.6071	0.7813	0.7500	0.4605	0.8472	0.6389	0.7895
$[30, 60] \times [20, 30]$	0.4643	0.3929	0.4063	0.3333	0.2763	0.4583	0.4028	0.3026
$[60, 120] \times [20, 30]$	0.1429	0.1429	0.3125	0.2083	0.2632	0.2222	0.2639	0.1711
$[> 120] \times [20, 30]$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$[0, 15] \times [> 30]$	0.3214	0.5000	0.5313	0.5833	0.7500	0.5833	0.5000	0.6842
$[15, 30] \times [> 30]$	0.4643	0.5000	0.3438	0.5000	0.5921	0.5278	0.4444	0.5658
$[30, 60] \times [> 30]$	0.2500	0.3571	0.4375	0.5417	0.3026	0.2778	0.2222	0.1842
$[60, 120] \times [> 30]$	0.0000	0.2857	0.0313	0.0000	0.1579	0.1389	0.2222	0.2368
$[> 120] \times [> 30]$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 3. Distribution of $P(A_x)$

$A_x = t_w \times t_s$	A	B	C	D
$[0, 15] \times [0, 10]$	0.2000	0.1481	0.1667	0.1111
$[15, 30] \times [0, 10]$	0.0000	0.0000	0.0000	0.0370
$[30, 60] \times [0, 10]$	0.4000	0.5556	0.1667	0.0370
$[60, 120] \times [0, 10]$	0.2000	0.0741	0.4583	0.4815
$[> 120] \times [0, 10]$	0.0800	0.0741	0.0000	0.0370
$[0, 15] \times [10, 15]$	0.0000	0.0000	0.0417	0.0370
$[15, 30] \times [10, 15]$	0.0000	0.0000	0.0000	0.0370
$[30, 60] \times [10, 15]$	0.0400	0.1111	0.0000	0.0370
$[60, 120] \times [10, 15]$	0.0400	0.0370	0.0417	0.0741
$[> 120] \times [10, 15]$	0.0000	0.0000	0.0000	0.0370
$[0, 15] \times [15, 20]$	0.0000	0.0000	0.0417	0.0000
$[15, 30] \times [15, 20]$	0.0000	0.0000	0.0000	0.0000
$[30, 60] \times [15, 20]$	0.0000	0.0000	0.0000	0.0370
$[60, 120] \times [15, 20]$	0.0400	0.0000	0.0833	0.0370
$[> 120] \times [15, 20]$	0.0000	0.0000	0.0000	0.0000
$[0, 15] \times [20, 30]$	0.0000	0.0000	0.0000	0.0000
$[> 15] \times [> 20]$	0.0000	0.0000	0.0000	0.0000

Table 4. Similarity

	A	B	C	D
Case (i)	0.5053	1.0000	0.7425	0.7974
Case (ii)	0.6385	0.4706	0.6226	0.8114

probability. The four values of the similarity $S(E^*_\varphi, F^*_\psi)$ for the cases (i) and (ii) are reported in Table 4. Notice that all the values are greater than 0.47 and that the greatest similarity ("on the average") is that of case (i): so from now on (due also to lack of space) we will consider only the case (i).

Similarity between pairs of services A, B, C, D can be considered to show that min-transitivity (as recalled at the end of Section 2) does not hold. A simple computation based on formula (1) gives, e.g.,

$$S(B, C) = 0.6952\ ,\ S(C, D) = 0.7681$$

(with an obvious meaning of the symbols), while $S(B, D) = 0.5191$, so that

$$S(B, D) < \min\{S(B, C)\ ,\ S(C, D)\}\ .$$

Finally, according to the number of doctors and patients involved, we can evaluate "calibrated" memberships $\mu_G(x)$, for the four services, by choosing the coefficient γ appearing in formula (5) equal, for example, to 1/5 (of course, different values could be obtained by different choices of the coefficient γ – depending on possible different judgments concerning the relevance of the opinions of doctors and patients, besides the trivial aspect concerning their numbers – but we

do not deem it useful to broaden our discussion on this). These values of $\mu_G(x)$ are inserted into formula (6), so we get an evaluation of the efficiency of the four services, that is

$$\mathcal{E}_A = 0.6465\,,\ \mathcal{E}_B = 0.6904\,,\ \mathcal{E}_C = 0.7151\,,\ \mathcal{E}_D = 0.6805\,.$$

References

1. Berwick, D.M.: Continuous improvement as an ideal in health care. The New England Journal of Medicine 320, 53–56 (1989)
2. Bouchon–Meunier, B., Rifqi, M., Bothorel, S.: Toward general measures of comparison of objects. Fuzzy Sets and Systems 84, 143–153 (1996)
3. Coletti, G., Scozzafava, R.: Characterization of Coherent Conditional Probabilities as a Tool for their Assessment and Extension. International Journal of Uncertainty, Fuzziness and Knowledge-Based System 4, 103–127 (1996)
4. Coletti, G., Scozzafava, R.: Conditioning and Inference in Intelligent Systems. Soft Computing 3, 118–130 (1999)
5. Coletti, G., Scozzafava, R.: The role of coherence in eliciting and handling "imprecise" probabilities and its application to medical diagnosis. Information Sciences 130, 41–65 (2000)
6. Coletti, G., Scozzafava, R.: Probabilistic logic in a coherent setting, Trends in Logic, vol. 15. Kluwer, Dordrecht (2002)
7. Coletti, G., Scozzafava, R.: Conditional probability, fuzzy sets and possibility: a unifying view. Fuzzy Sets and Systems 144, 227–249 (2004)
8. Coletti, G., Scozzafava, R.: Conditional Probability and Fuzzy Information. Computational Statistics & Data Analysis 51, 115–132 (2006)
9. De Baets, B., De Meyer, H.: Transitivity-preserving fuzzification schemes for cardinality-based similarity measures. European Journal of Operational Research 160, 726–740 (2005)
10. de Finetti, B.: Sull'impostazione assiomatica del calcolo delle probabilità. Annali Univ. Trieste 19, 3–55 (1949) Engl. transl. In: Probability, Induction, Statistics, ch. 5, Wiley, London (1972)
11. de Finetti, B.: Teoria della probabilità, Einaudi, Torino (1970) - Engl. transl.: Theory of Probability, vol. 1& 2., Wiley, Chichester (1974)
12. Donabedian, A.: An Introduction to Quality Assurance in Health Care. Oxford University Press, Oxford (2003)
13. Gaynord, C.J.: Monitoring with indicators: evaluating the quality of patient care. Aspen Publication, Gaithersburg, Maryland (1998)
14. Kachukhashvili, G.S., Tsiskarishvili, N.E., Dubovik, M.V., Badiashvili, G.V.: The use of fuzzy sets techniques in managing health organizations. Med. Info. 8 (1995)
15. Nelson, E.C., Slaine, M.E., Batalden, P.B., Plume, S.K.: Building measurement and data collection into medical practice. Annals of Internal Medicine 128, 460–466 (1998)
16. Powell, A., Davies, H., Thompson, R.: Using routine comparative data to assess quality of health care. Quality and Safety in Health Care 12, 122–128 (2003)
17. Scozzafava, R., Vantaggi, B.: Fuzzy relations in a coherent conditional probability setting. In: 7th International Conference on Information and Management Sciences (IMS), Chengdu, China, pp. 496–500 (2006)
18. Zadeh, L.A.: Toward a perception–based theory of probabilistic reasoning with imprecise probabilities. Journal of Statistical Planning and Inference 105, 233–264 (2002)

Automatic Indexing from a Thesaurus Using Bayesian Networks: Application to the Classification of Parliamentary Initiatives

Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, and Alfonso E. Romero

Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S.I. Informática y de Telecomunicaciones, Universidad de Granada
18071 – Granada, Spain
{lci,jmfluna,jhg,aeromero}@decsai.ugr.es

Abstract. We propose a method which, given a document to be classified, automatically generates an ordered set of appropriate descriptors extracted from a thesaurus. The method creates a Bayesian network to model the thesaurus and uses probabilistic inference to select the set of descriptors having high posterior probability of being relevant given the available evidence (the document to be classified). We apply the method to the classification of parliamentary initiatives in the regional Parliament of Andalucía at Spain from the Eurovoc thesaurus.

1 Introduction

To improve organizational aspects and facilitate fast access to relevant information relative to a particular subject, document collections from many organizations are classified according to their content using a set of descriptors extracted from some kind of controlled vocabulary or thesaurus. For example, most of the parliaments in Europe use a thesaurus called Eurovoc to classify parliamentary initiatives, the Food and Agricultural Organization (FAO) employs Agrovoc to categorize its documents, and the National Library of Medicine (NLM) uses MeSH to index articles from biomedical journals. The process of assigning descriptors in the thesaurus to the documents is almost always carried out manually by a team of expert documentalists. The objective of this work is the development of a computerized tool to assist the human experts in this process.

So, the scope of the paper is automatic subject indexing from a controlled vocabulary [6,10] and hierarchical text classification [11,14]. However, given the critical nature of this task in many contexts, it is not realistic to try to design a completely automatic classification process, and final human supervision will always be required.

An important characteristic of the model that we are going to propose is that no training is required. We shall exploit only the hierarchical and equivalence relationships among the descriptors in the thesaurus. This is an advantage because the model may be used with almost any thesaurus and without having

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 865–877, 2007.

preclassified documents (in a large hierarchy, the amount of preclassified document necessary for training may be huge). On the other hand, this is also a weakness because any kind of information not considered in the thesaurus (e.g. other synonymy relations, specific information handled by documentalists,...) will not be taken into account. Consequently, we cannot expect very high rates of success in comparison with classifiers that are built starting from training data [3,5,9,13]. In this sense our proposal is more similar to the work in [1,2], where a method to populate an initially empty taxonomy is proposed. The working hypothesis is that a documentalist would prefer to confirm or discard a given classification hypothesis proposed by the system rather than examining all the possible alternatives.

Another important characteristic of our model is that is based on Bayesian networks. To the best of our knowledge, no Bayesian network-based models other than naive Bayes have been proposed to deal with this kind of problems [7]. We create a Bayesian network to model the hierarchical and equivalence relationships in the thesaurus. Then, given a document to classify, its terms are instantiated in the network and a probabilistic inference algorithm computes the posterior probabilities of the descriptors in the thesaurus.

In Section 2 we describe the proposed Bayesian network model of a thesaurus. The experimental evaluation is explained in Section 3. Finally, Section 4 contains the final remarks and some proposals for future work.

2 The Bayesian Network Model of a Thesaurus

In this section we shall first describe the general structure of a thesaurus and next the basic Bayesian network model that we propose to represent it, including the graphical structure, the conditional probabilities, the inference mechanism and some implementation details, and later a possible improvement. We assume that the reader has at least a basic background on Bayesian networks [12].

2.1 Thesaurus Structure

Any thesaurus comprises *descriptors* or *indexing terms*, *non-descriptors* or *entry terms* and *semantic relationships*, which may be equivalence, hierarchical and associative relationships. Descriptors are words or expressions which denote in unambiguous fashion the constituent concepts of the field covered by the thesaurus, whereas non-descriptors are words or expressions which in natural language denote the same or a more or less equivalent concept as a descriptor in the language of the thesaurus.

The *equivalence relationship* between descriptors and non-descriptors in fact covers relationships of several types: genuine synonymy, near-synonymy, antonymy and inclusion, when a descriptor embraces one or more specific concepts which are given the status of non-descriptors because they are not often used. It is usually represented by the abbreviations "UF" (Used For), between the descriptor and the non-descriptor(s) it represents, and "USE" between a non-descriptor

and the descriptor which takes its place. The *hierarchical relationship* between descriptors is shown by the abbreviations: "NT" (Narrower Term) between a generic descriptor and a more specific descriptor, and "BT" (Broader Term) between a specific descriptor and a more generic descriptor. Descriptors which do not contain other more specific descriptors are called *basic descriptors*; otherwise they are called *complex descriptors*. Descriptors which are not contained in any other broader descriptors are *top descriptors*. Sometimes a few descriptors may be polyhierarchical (they have more than one broader descriptor). This means that the hierarchical relationships do not form a tree but a graph. The *associative relationship*, shown by the abbreviation "RT" (Related Term), relates two descriptors that do not meet the criteria for an equivalence nor a hierarchical relationship. It is used to suggest another descriptor that would be helpful for the thesaurus user to search by. In this work we shall not consider associative relationships.

Example. Eurovoc is a multilingual thesaurus covering the fields in which the European Communities are active. Figure 1 displays the BT relationships between some descriptors of Eurovoc and the USE relationships between the non-descriptors and these descriptors. There are two complex descriptors, *abortion* and *birth control*, and four basic descriptors, *illegal abortion*, *therapeutic abortion*, *contraception* and *sterilisation*. The associated non-descriptors are: *legal abortion*, *termination of pregnancy* and *voluntary termination of pregnancy* for *abortion*; *birth spacing* for *birth control*; and *tubal ligation* and *vasectomy* for *sterilisation*.

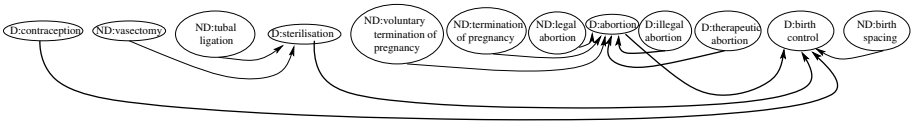


Fig. 1. BT (bold lines) and USE (normal lines) relationships for the descriptors and non-descriptors in the example about *abortion*

2.2 Basic Network Structure

In order to develop a Bayesian network (BN) for modeling a thesaurus, a naive approach would be to use a type of representation as the one in Fig. 1, containing descriptor and non-descriptor nodes, then adding term nodes representing the words in the thesaurus and connecting them with the descriptor and non-descriptor nodes that contain these words. This would result in a network structure as the one displayed in Fig. 2. The problem with this type of topology is that each descriptor node receives two or three kinds of arcs with different meaning (those from its non-descriptor nodes and those from its term nodes in the case of basic descriptor nodes and, for the case of complex descriptor nodes, also those arcs from the narrower descriptor nodes that they contain). As this would make

much more difficult the process of assigning the associated conditional probability distributions to the nodes, we propose a different topology. The idea is to distinguish between a concept and the descriptor and non-descriptors used to represent it.

Each concept, labeled identically as the descriptor representing it, will be a node in the network. We shall also distinguish between basic and complex concepts: the former do not contain other concepts, whereas the later are composed of other concepts (either basic or complex). Each descriptor and each non-descriptor in the thesaurus will also be nodes in the network. All the words or terms appearing in either a descriptor or a non-descriptor will be term nodes.

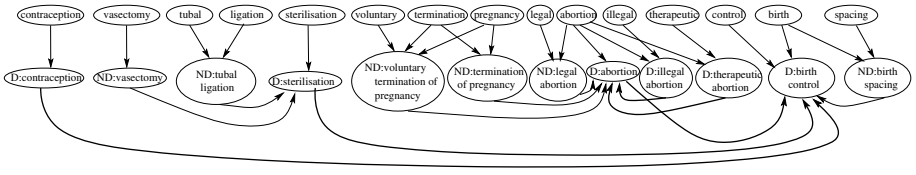


Fig. 2. Preliminary Bayesian network in the example about *abortion*

There is an arc from each term node to each descriptor and/or non-descriptor node containing it. There are also arcs from each non-descriptor node to the associated concept node (these arcs correspond with the USE relationships), as well as from the descriptor node representing the concept to the concept node itself.

As the complex concepts, in addition to its own specific information (descriptors and non-descriptors), are also containers of other concepts, for each complex concept we shall also create a duplicate (virtual) descriptor node which will receive the influence of the concepts contained in the complex concept. Therefore, there is an arc from each concept node which is not associated with a top descriptor to the virtual descriptor node associated with the broader complex concept(s) containing it (these arcs correspond with the BT relationships), as well as an arc going from each virtual descriptor node to its corresponding complex concept node.

We shall denote $\mathcal{T}$ the set of term nodes, $\mathcal{DE}$ and $\mathcal{ND}$ the sets of descriptor and non-descriptor nodes, respectively, $\mathcal{C}$ the set of concept nodes and $\mathcal{V}$ the set of virtual descriptor nodes. All the nodes will represent binary random variables. The domain of each variable is: $\{t^+, t^-\}\ \forall T \in \mathcal{T}$; $\{de^+, de^-\}\ \forall DE \in \mathcal{DE}$; $\{nd^+, nd^-\}\ \forall ND \in \mathcal{ND}$; $\{c^+, c^-\}\ \forall C \in \mathcal{C}$; $\{v^+, v^-\}\ \forall V \in \mathcal{V}$. For term nodes, their values indicate whether the term appear in the document to be classified. For descriptor and non-descriptor nodes, the values represent whether the corresponding descriptor or non-descriptor may be associated with the document. For concept nodes the values mean whether the concept is appropriate/relevant to classify the document. $Pa(X)$ will represent the parent set of a node X in the graph. The network topology that we are proposing is completely determined by specifying the parent set of each node: for each term node $T \in \mathcal{T}$, $Pa(T)$

is the empty set; for each descriptor and non-descriptor node $DE \in \mathcal{DE}$ and $ND \in \mathcal{ND}$, $Pa(DE)$ and $Pa(ND)$ are in both cases the set of term nodes associated with the words that appear in DE and ND, respectively; for each concept node $C \in \mathcal{C}$, $Pa(C)$ is the set of descriptor and non-descriptor nodes that define the concept and, in the case of complex concept nodes, also its associated virtual descriptor node, V_C; finally, for each virtual descriptor node $V \in \mathcal{V}$, $Pa(V)$ is the set of concept nodes (either basic or complex) contained in the corresponding complex concept.

For the previous example the corresponding subnetwork is shown in Fig. 3. The nodes labeled with D and ND are descriptor and non-descriptor nodes, respectively. The nodes labeled with C are concept nodes and those labeled with V are virtual descriptor nodes. The remaining nodes are term nodes.

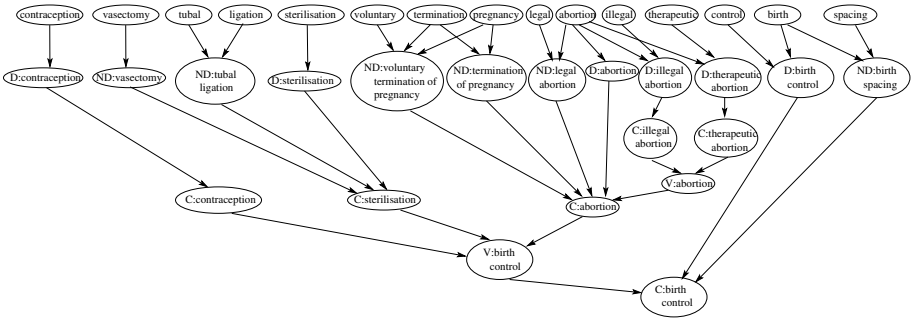


Fig. 3. Bayesian network in the example about *abortion*

2.3 Types of Conditional Probability Distributions

The probability distributions that must be specified are the prior probabilities for term nodes, $p(t^+)$, and the following conditional probabilities: for descriptor and non-descriptor nodes, $p(de^+|pa(DE))$ and $p(nd^+|pa(ND))$ respectively, for concept nodes, $p(c^+|pa(C))$, and for virtual descriptor nodes, $p(v^+|pa(V))$. In all the cases $pa(X)$ represents a configuration of the parent set $Pa(X)$ of the node X.

For the prior probabilities of term nodes we propose using a constant value, $p(t^+) = p_0$, $\forall T \in \mathcal{T}$ (although we shall see later that this is not an important issue at all).

As the treatment of the descriptor and non-descriptor nodes will the same, in order to simplify the exposition, from now on we will denote $\mathcal{D} = \mathcal{DE} \cup \mathcal{ND}$ and we will refer to both descriptor and non-descriptor nodes as descriptor nodes. An element in $\mathcal{D}$ will be denoted as D. For the conditional probabilities of a descriptor node D given the terms that it contains, $p(d^+|pa(D))$, we propose using a canonical additive model [4], employed in the information retrieval field:

$$\forall D \in \mathcal{D},\ p(d^+|pa(D)) = \sum_{T \in R(pa(D))} w(T, D)\ , \tag{1}$$

where $w(T, D)$ is the weight associated to each term T belonging to the descriptor D. $R(pa(D))$ is the subset of parents of D which are observed in the configuration $pa(D)$, i.e., $R(pa(D)) = \{T \in Pa(D) \,|\, t^+ \in pa(D)\}$. So, the more parents of D are observed the greater its probability of relevance. These weights can be defined in any way, the only restrictions are that $w(T, D) \geq 0$ and $\sum_{T \in Pa(D)} w(T, D) \leq 1$.

For the conditional probabilities of each concept node C given the descriptor nodes that define the concept and its virtual descriptor node (in the case of complex concept nodes), $p(c^+|pa(C))$, it is not appropriate to use the previous additive model, because each descriptor alone is supposed to be able to represent the concept, and this behaviour cannot be obtained using an additive model. So, we propose to use another kind of canonical model, namely an OR gate [12]:

$$\forall C \in \mathcal{C},\ p(c^+|pa(C)) = 1 - \prod_{D \in R(pa(C))} (1 - w(D, C)) \quad . \tag{2}$$

$R(pa(C)) = \{D \in Pa(C) \,|\, d^+ \in pa(C)\}$ and $w(D, C)$ is the probability that the descriptor D alone (the other descriptors being non relevant) makes concept C relevant, with $0 \leq w(D, C) \leq 1$.

For the conditional probabilities of each virtual descriptor node V given the concept nodes it comprises, $p(v^+|pa(V))$, we can use again the previous additive canonical model, because the more relevant are all the concepts contained in the complex concept associated to V, the more clearly this broader concept is appropriate:

$$\forall V \in \mathcal{V},\ p(v^+|pa(V)) = \sum_{C \in R(pa(V))} w(C, V) \quad . \tag{3}$$

$R(pa(V)) = \{C \in Pa(V) \,|\, c^+ \in pa(V)\}$ and $w(C, V)$ is the weight of the concept C in V, with $w(C, V) \geq 0$ and $\sum_{C \in Pa(V)} w(C, V) \leq 1$.

2.4 Quantifying the Conditional Probabilities

To define the weight of a term in a descriptor, $w(T, D)$, we propose a normalized tf-idf scheme:

$$w(T, D) = \frac{tf(T, D) * idf(T)}{\sum_{T' \in Pa(D)} tf(T', D) * idf(T')} \quad .$$

The *inverse descriptor frequency* of a term, $idf(T)$, is

$$idf(T) = \ln\left(\frac{m}{n(T)}\right) \quad ,$$

where $n(T)$ is the number of descriptors and non-descriptors in the thesaurus that contain the term T and m is the total number of descriptors and non-descriptors. The *term frequency* of a term in a descriptor, $tf(T, D)$, is the number of times that this term appears in the descriptor (which will be almost always equal to 1, because the descriptors usually contain very few words).

For the weights of the descriptors in the concepts, $w(D, C)$, a reasonable choice is a value near 1.0, because any descriptor associated with a concept represents it perfectly (descriptors and non-descriptors associated with a concept are assumed to be synonymous in the language of the thesaurus). In the experiments we have used $w(D, C) = 0.9$, in order to discriminate between concepts having a different number of descriptors that match with the document to be classified.

Finally, for the weights of the component concepts in each virtual descriptor, $w(C, V)$, we propose to use uniform weights (there is no reason to believe that a concept is more important than another one with respect to the broader concept containing them). Therefore:

$$w(C, V) = \frac{1}{|Pa(V)|} \ .$$

2.5 Inference

Given a document Q to be classified/indexed, the process is first to instantiate in the network the term nodes corresponding to the words appearing in Q as observed and the remaining term nodes as not observed. Let q be such a configuration of the term nodes in $\mathcal{T}$. Then we propagate this information through the network and compute the posterior probabilities of the concept nodes, $p(c^+|q)$. Finally, the descriptors associated with the concept nodes having greater posterior probability are used to classify the document.

To compute the posterior probabilities of the concept nodes, we can take advantage of both the network topology and the canonical models being considered. As all the term nodes are instantiated to either observed or non-observed, then all the descriptor nodes which are parents of a concept (including the associated virtual descriptor if it exists) are conditionally independent given q. In this case, taking into account that the canonical model for the concept nodes is an OR gate, we can compute these probabilities as follows [12]:

$$p(c^+|q) = 1 - \prod_{D \in Pa(C)} \left(1 - w(D, C)p(d^+|q)\right) \ .$$

As the weights $w(D, C)$ are all equal to 0.9, we have:

$$p(c^+|q) = 1 - \prod_{D \in Pa(C)} \left(1 - 0.9p(d^+|q)\right) \ . \tag{4}$$

The probabilities of the (non virtual) descriptor nodes can be calculated, according to the additive model being used, as follows [4]:

$$p(d^+|q) = \sum_{T \in Pa(D)} w(T, D)p(t^+|q) \ .$$

As $p(t^+|q) = 1 \ \forall T \in Pa(D) \cap Q$ and $p(t^+|q) = 0 \ \forall T \in Pa(D) \setminus Q$, we obtain:

$$p(d^+|q) = \sum_{T \in Pa(D) \cap Q} w(T, D) \ . \tag{5}$$

The computation of the posterior probabilities of the virtual descriptor nodes is also very simple, using again the properties of the additive canonical model considered:

$$p(v^+|q) = \frac{1}{|Pa(V)|} \sum_{C \in Pa(V)} p(c^+|q) \ . \tag{6}$$

This computation can be carried out as soon as the posterior probabilities of all the concept nodes included in V are known.

Therefore, we compute first the posterior probabilities of all the descriptor nodes using (5), then the posterior probabilities of the basic concept nodes (which have no virtual descriptor) using (4). Next, we can compute in a top-down manner the posterior probabilities of the virtual descriptor nodes and the complex concept nodes using (6) and (4), respectively.

2.6 Implementing the Model

In this section we shall study in more detail how to implement in an efficient way the proposed model. We start from the term nodes associated with the words appearing in the document to be classified. For each one of them, we accumulate the weights of these term nodes in the descriptor nodes containing them. After this process, each visited descriptor node D contains the value $v[D] = \sum_{T \in Pa(D) \cap Q} w(T, D)$, i.e. $p(d^+|q)$, according to (5) (the posterior probability of the non visited descriptor nodes is equal to zero).

Next, starting from each of the visited descriptor nodes, we visit the concept node containing it and compute the product $\prod_{D \in Pa(C)} (1 - 0.9v[D])$ progressively. After this step each visited basic concept node contains, according to (4), the value $v[C] = 1 - p(c^+|q)$ (the non visited basic concept nodes have a posterior probability equal to zero) and each visited complex concept node contains the value $v[C] = (1 - p(c^+|q))/(1 - 0.9p(v_c^+|q))$, because the contribution of its virtual descriptor node has not been computed yet.

Finally, we traverse the subgraph induced by the set of visited concept nodes and their descendants in a topological ordering (parents before children). If the visited node is a basic concept node C, we directly compute $p(c^+|q)$ (by setting $v[C] = 1 - v[C]$). If the visited node is a virtual node V, we compute its probability by adding the values already computed for its parent concept nodes and dividing by the number of parents, according to (6). If the visited node is a complex concept node C, we compute its probability by subtracting from 1 the value obtained by multiplying its stored value and the value already computed for its associated virtual node, $v[C] = 1 - v[C](1 - 0.9v[V_C])$. It can be easily seen that the complexity of this process is linear in the number of arcs in the graph[1]. It is worth mentioning that the actual implementation manages the BN implicitly, i.e. the Bayesian network is never explicitly constructed; instead, we directly use the BT, NT and USE relationships in the thesaurus, augmented

[1] More precisely, the complexity is linear in the number of arcs of the subgraph induced by the term nodes appearing in the document Q and their descendant nodes.

with two inverted file-like structures to store, for each word in the thesaurus, the lists of descriptors and non-descriptors that contain it.

2.7 Taking Degree of Coverage into Account

There is another dimension of the concepts in a thesaurus with respect to the document to be classified that we have not considered yet. We call this property the *coverage* of a concept with respect to a document, which tries to discriminate between concepts which are almost surely relevant for the document: if two concepts are initially considered equally relevant to classify a document but one of them includes more descriptors appearing in the document than the other, the former should be preferable. This strategy is motivated by the common guidelines being used to manually classify documents: we should use the most specific concepts available to bring out the main focus of a document and, if the document covers several specific concepts, then we should use as many specific concepts from different subtrees as required by the content of the document. *However, when several specific concepts are needed that fall within the same subtree structure, the broader concept should be assigned instead.*

Using the previous Bayesian network model, if, for instance, the three concepts which are included into a broader concept are completely relevant for a given document, then this broader concept also becomes completely relevant and therefore the four concepts would be (wrongly) assigned to the document.

To overcome this problem, we shall define the coverage of a concept C, $cov(C)$, as the set of concepts which are ancestors of C in the Bayesian network, together with C itself, i.e. all the concepts which are specializations (at different levels of granularity) of C. For example, the coverage of the concept *birth control* are the concepts *abortion*, *contraception*, *sterilisation*, *illegal abortion*, *therapeutic abortion* and *birth control*. Roughly speaking, the degree of coverage of a concept with respect to a document is the proportion of the document which is within the coverage of the concept. More concretely, $\forall C \in \mathcal{C}$, let us define $An_t(C) = \{T \in \mathcal{T} \,|\, \exists B \in cov(C), \exists D \in Pa(B) \text{ and } T \in Pa(D)\}$. In words, $An_t(C)$ is the set of terms in the thesaurus which are part of a descriptor associated to a concept in the coverage of C. We formally define the degree of coverage of a concept C with respect to a document Q, $dc(C, Q)$, as:

$$dc(C, Q) = \frac{\sum_{T \in An_t(C) \cap Q} idf(T)}{\sum_{T \in Q} idf(T)} \quad .$$

The decision about what descriptors to assign to a document should be made, not only depending on the probability of relevance of the concepts but also in terms of the degree of coverage of these concepts.

In order to formally include these ideas in the model, we shall think in terms of Decision theory, by defining a utility function based on the degree of coverage and then computing the expected utility of assigning a concept to a document. Those concepts having higher expected utility will be used to classify the document. If we define the utility of assigning the concept C to the document Q when

C is truly relevant as $dc(C, Q)$, and the utility of assigning C to Q when C is not relevant as zero, then the expected utility of assigning C to Q is simply $p(c^+|q) \times dc(C, Q)$.

3 Experimental Evaluation

Our experiments have been carried out using a data base provided by the Parliament of Andalucía at Spain, containing 7933 parliamentary initiatives manually classified using descriptors from an adapted version of the Eurovoc thesaurus. This version contains 5080 descriptors, 6975 non-descriptors and 7120 distinct words (excluding stopwords)[2]. The average number of assigned descriptors per initiative is 3.8. We have not used the full text of the initiatives but only a short summary (typically two or three lines of text). As our aim is not a complete but only a partial automation of the classification process, the selected performance measures have been the *average recall-precision curve* and the *average 11-point precision*[3], which are frequently used for category-ranking classifiers [14].

We have experimented with two alternatives: (1) the basic Bayesian network alone (BN) and (2) using coverage (BN+C). Moreover, each of these options has been tested with and without using stemming, although we always use stopword removal. The recall-precision curves of the four alternatives are displayed in Fig. 4, whereas the average 11-point precision values are shown in Table 1. With respect to the efficiency of the inference process, all the 7933 initiatives were classified in around 10 seconds on a computer equipped with an Intel Core2 duo 2GHz processor.

In order to assess the quality of the proposed BN-based models, we have also experimentally compared them with two simple benchmark methods. The first one [8,15] ranks concepts for a document based on word matching between the document and the lexical information associated to the concepts in the thesaurus, using a conventional vector space model (VSM) and the cosine measure: each document to be classified is considered as a query against a "document collection" where each "document", representing a concept, is indexed using the words appearing in the descriptor and non-descriptors which are associated with the concept. This approach uses only the lexical information, while topological (hierarchical) information is neglected. A second approach which also exploits the hierarchical information (HVSM) [1,2] is based on the idea that the meaning of a concept in the thesaurus is a specialization of the meaning of the broader concepts containing it[4]. Therefore, all the words appearing in the descriptors and non-descriptors of the broader concepts of a given concept are also used to index the "document" associated with this concept. The results obtained by

[2] The BN representing the thesaurus contains more than 25000 nodes.

[3] The precision values are interpolated at 11 points at which the recall values are 0.0, 0.1,. . ., 1.0, and then averaged.

[4] In the language of our Bayesian network model, these broader concepts would be the descendants of the concept being considered.

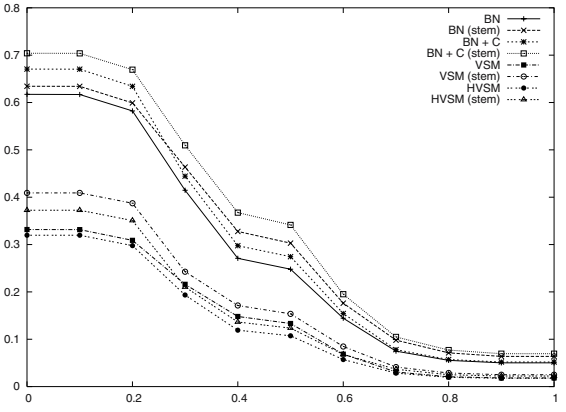


Fig. 4. Average recall-precision curves

Table 1. Average 11-point precision for the different experiments

Using stemming				Without using stemming			
BN+s	BN+C+s	VSM+s	HVSM+s	BN	BN+C	VSM	HVSM
0.3123	0.3466	0.1798	0.1582	0.2841	0.3123	0.1478	0.1361

these two benchmark models, once again with and without using stemming, are also displayed in Fig. 4 and Table 1.

Several conclusions may be obtained from these experiments: first, as the BN-based models always provide much better results than both the simple and hierarchical vector space models, it seems that the Bayesian network approach is useful in this classification problem. Second, stemming is also recommendable in this context, because its use always improves the results. Third, using coverage is clearly advantageous. Fourth, concerning the vector space model, in this case the use of the hierarchical information is self-defeating and produces results worse than those of the simple VSM[5]. Finally, the model performance is in general quite acceptable, specially at lower points of recall, reaching a precision near 70%.

4 Concluding Remarks

We have developed a Bayesian network-based model for hierarchical classification of documents from a thesaurus. The experimental results obtained using a large set of parliamentary initiatives from the Parliament of Andalucía and the Eurovoc thesaurus are encouraging, specially if we consider that no training data

[5] This contrasts with the results obtained in [1] in the context of hierarchical classification of documents into web directories, where the hierarchical VSM generally outperformed the simple VSM.

are used to build the model, and outperform those of the two simple benchmark methods considered.

For future research, we are planning to improve the model in three different ways: first, by considering the *context* of the terms/descriptors appearing in a document. The idea is to avoid assigning to a document a descriptor whose appearance may be incidental or their meaning within the document being quite different from the intended meaning within the thesaurus. Second, by taking also into account the associative relationships between descriptors in the thesaurus. Third, by integrating the model within a more general scheme where training data, in the form of preclassified documents, may also be used.

Acknowledgments

This work has been supported by the Spanish 'Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía' and 'Ministerio de Educación y Ciencia', under Projects TIC-276 and TIN2005-02516 respectively.

References

1. Adami, G., Avesani, P., Sona, D.: Clustering documents in a web directory. In: Proceedings of Fifth ACM Int. Workshop on Web Information and Data Management, pp. 66–73. ACM Press, New York (2003)
2. Adami, G., Avesani, P., Sona, D.: Clustering documents into a web directory for bootstrapping a supervised classification. Data & Knowledge Engineering 54, 301–325 (2006)
3. Chakrabarti, S., Dom, B., Agrawal, R., Raghavan, P.: Using taxonomy, discriminants, and signatures for navigating in text databases. In: Proceedings of the 23rd International Conference on Very Large Data Bases, pp. 446–455 (1997)
4. de Campos, L.M., Fernández-Luna, J.M, Huete, J.F.: The BNR model: foundations and performance of a Bayesian network-based retrieval model. International Journal of Approximate Reasoning 34, 265–285 (2003)
5. Dumais, S., Chen, H.: Hierarchical classification of web document. In: Proceedings of the 23th ACM International Conference on Research and Development in Information Retrieval, pp. 256–263. ACM Press, New York (2000)
6. Golub, K.: Automated subject classification of textual web documents. Journal of Documentation 62(3), 350–371 (2006)
7. Koller, D., Sahami, M.: Hierarchically classifying documents using very few words. In: Proceedings of the 14th International Conference on Machine Learning, pp. 170–178 (1997)
8. Larson, R.R.: Experiments in automatic library of congress classification. Journal of the American Society for Information Science 43(2), 130–148 (1992)
9. Lauser, B., Hotho, A.: Automatic multi-label subject indexing in a multilingual envirinment. In: Koch, T., Sølvberg, I.T. (eds.) ECDL 2003. LNCS, vol. 2769, pp. 140–151. Springer, Heidelberg (2003)
10. Medelyan, O., Witten, I.: Thesaurus based automatic keyphrase indexing. In: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries, pp. 296–297 (2006)

11. Moskovitch, R., Cohen-Kashi, S., Dror, U., Levy, I.: Multiple hierarchical classification of free-text clinical guidelines. Artificial Intelligence in Medicine 37(3), 177–190 (2006)
12. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan and Kaufmann, San Mateo (1988)
13. Ruiz, M., Srinivasan, P.: Hierarchical text categorization using neural networks. Information Retrieval 5(1), 87–118 (2002)
14. Sebastiani, F.: Machine Learning in automated text categorization. ACM Computing Surveys 34, 1–47 (2002)
15. Yang, Y.: An evaluation of statistical approaches to text categorization. Information Retrieval 1, 69–90 (1999)

A Genetic Programming Classifier Design Approach for Cell Images

Aydın Akyol, Yusuf Yaslan, and Osman Kaan Erol

Computer Engineering Department, Istanbul Technical University, Turkey
{aydin.akyol, yyaslan,okerol}@itu.edu.tr

Abstract. This paper describes an approach for the use of genetic programming (GP) in classification problems and it is evaluated on the automatic classification problem of pollen cell images. In this work, a new reproduction scheme and a new fitness evaluation scheme are proposed as advanced techniques for GP classification applications. Also an effective set of pollen cell image features is defined for cell images. Experiments were performed on Bangor/Aberystwyth Pollen Image Database and the algorithm is evaluated on challenging test configurations. We reached at 96 % success rate on the average together with significant improvement in the speed of convergence.

Keywords: Genetic programming, cell classification, classifier design, pollen classification.

1 Introduction

Genetic programming (GP) is an automatic programming approach which is relatively recent and fast developing, especially to solve optimization problems. The solutions to the problem are represented as computer programs which are obtained by using Darwin's principle of natural selection and recombination. In short, GP becomes a powerful method to solve the NP hard problems by using the capabilities of evolutionary search [1].

As in [3,4,5,6] GP was applied successfully to many real-world applications. Some of these approaches use dynamic class labels. Early works for GP-classification employed only simple mathematical expressions to embed in the result programs. Then in addition to these simple expressions, conditional operators, arithmetic operators, high-level mathematical operators and some heuristics were also used as in [2].

Automatic labeling of pollens is not a new problem since pollen cells play a major role in many areas, such as study of allergic reactions in medicine [7] and palaeo-environmental reconstruction [8]. Traditionally, the classification problem of pollen cells is treated as a multi-class problem and it is solved by applying minimum distance classifier either in pixel domain or on some sub-space. Also in [9] neural network models are proposed for the problem. Some of the features used in common are coefficients obtained from co-occurrence matrices, gray level run length statistics, measurements obtained from neighboring dependencies and simple first order

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 878–888, 2007.

statistics. In [7] Rodriguez et. al. reached to %90 success rate by using similar shape and brightness features.

We propose a new GP classification strategy for binary pollen cell classification problems. The basic idea is the conversion of the numeric output of a genetic program classifier into class labels by using new fitness evaluation and reproduction schemes based on BigBang-BigCrunch method [10]. In reproduction operation, one individual of the current population is selected according to its fitness value; the better the fitness value of an individual the higher the probability to be selected for the next generation. BB-BC algorithm has two important stages; first stage (Big Bang) creates the initial population randomly and the candidate solutions are spread all over the search space in a uniform manner, and the next stage (Big Crunch) is a convergence operator that the output is derived by calculating the center of mass. New candidates are calculated around the center of mass by adding or subtracting normally distributed random numbers whose values are decreased at each of the iterations. The algorithm is iterated until a certain criterion is accomplished. Section 2.4 provides more detailed information about the BB-BC method based reproduction scheme. We obtain %96 success rate on average with this new GP classifier design.

Organization of the paper is as follows; Section 2 describes the overall design of the proposed GP algorithm and provides an analysis about using GP as a classifier. The process of the feature extraction is defined in Section 3. Section 4 presents the experimental setup and finally in Section 5 and 6, the results and the concluding remarks are given.

2 Genetic Programming Based Classification

In this approach, we use a tree-based structure to represent genetic programs. The ramped half-and-half method is employed to generate initial programs [1]. For selection, the proportional selection mechanism is employed and traditional one point cross-over is used for reproduction together with point-mutation.

In this section we address the other aspects of the GP learning/evolutionary system: selection of the terminal set, selection of the function set, construction of the fitness measure and selection of the input parameters and determination of the termination strategy.

2.1 Terminal Set

The terminals used in this application can be sub-divided into two groups; Image features and Real constants.

In this work, we use statistical features in various orders to build the variable terminal set. Although the use of domain-specific knowledge may be beneficial, we don't prefer to use any considering the computational burden of hand-crafted processes and increase in data-dependency. Section 3 provides a detailed description for the features used in this work.

We have also used a set of constant positive real numbers. The number of the constant terminals in a tree of an individual is determined probabilistically in run-time by considering a limit value. Constants are sampled from a uniform distribution.

2.2 Function Set

The function set and terminal set must have closure property: which means that each function must accept input values and return output values which belong to the interval defined by terminal set. As in [1] traditionally exceptions in mathematical functions are avoided by defining protected functions that are the constrained version of the original formulation.

We use three types of function sets; Arithmetic operations (**+,-,*,÷** (protected)), Mathematical functions (sin,cos), and Conditional functions (min,max,if).

The conditional functions min and max take two real values and compares them and return the selected value. The other conditional operator "if" takes four arguments. First two are used for the conditional part of the 'if' expression. The other two are the results to be returned according to the result of the comparison. The "if" function allows a program to contain various expressions in different regions of the feature space, thus allows discontinuous programs, rather than insisting on continuous or smooth functions. Furthermore, since the algorithm is used for binary classes, it can be expected that "min" and "max" functions may appear frequently in the resultant programs since the binary class separation would consist of many decisions.

2.3 Fitness Function

GP is guided by the fitness function to search for the most efficient computer program to solve a given problem. To calculate the classification accuracy of a genetic program, one needs to determine how to translate the program output into a class label. Because of the evolved genetic program has a numeric output value, it needs to be translated into class labels. For binary classification scheme this class labels are determined by 1's and 0's referring 'found' and 'not-found' respectively. To avoid slow convergence, in GP these labels should be determined by considering the design parameters, especially the range of constants used.

GP tries to find a nonlinear classifier for the input data in a supervised scheme. In theory, it would asymptotically reach to the exact classifier with the assumption of having unlimited resource (space and time) and using only Boolean logic operators. However, obtaining complex functions by using only Boolean operators would require great number of iterations and huge individual representations, so in practice high level functions are used directly. As it can be expected, this would bring some constraints in the search space. Also, it is quite probable that the GP algorithm would spend too much time to be able to relieve the negative effects of using such kind of high level functions.

In classification, a set of feature is mapped to a class label. First class labels and variables are encoded as real numbers and then the the classifier is expected to solve these M non-linear equations;

$$\begin{aligned} w_1.x_1 \,\Theta\, w_2.x_2 \,\Theta\, w_3.x_3 \,\Theta \ldots \Theta\, w_N.x_N &= C_1 \\ w_1.x_1 \,\Theta\, w_2.x_2 \,\Theta\, w_3.x_3 \,\Theta \ldots \Theta\, w_N.x_N &= C_2 \\ \ldots\ldots & \\ w_1.x_1 \,\Theta\, w_2.x_2 \,\Theta\, w_3.x_3 \,\Theta \ldots \Theta\, w_N.x_N &= C_M \end{aligned} \tag{1}$$

where w_i's stand for weights, x_i's are the features, Θ's are the function expressions and C_i's are class labels.

Asymptotically the exact weight expressions of the features and the function expressions would be found, but in practice only an approximation can be obtained. Thus it would be more justifiable to modify the representation of individuals given in Eq.1 as a nonlinear function;

$$f(x_1, x_2, ..., x_N, C) < \varepsilon \tag{2}$$

After from this brief analysis of GP as a classifier, now we can review the goal of the search process of GP as finding the optimum *f()* that will be valid for all training samples with minimum error. In the literature all GP classifiers use the number of matches as the measure of the quality of the individual [1-12]. In our design, we modify this evaluation criterion as the total distance of the inner class samples; thus, at the end of the training, we expect to find a discriminant which represents all these inner class points with minimum error. The fitness of an individual is computed as in Eq.3;

$$fitness(I) = \frac{1}{\sum |errors(I)| + \eta} \tag{3}$$

where I is the individual in question and ε denotes a pre-defined constant.

2.4 Reproduction Scheme

Reproduction scheme is determined according to the characteristics of the problem and the design issues of the GP algorithm. Generally elitism works with satisfactory performance for classification purposes. Although it gives satisfactory results, we propose an improved version of the Big Bang-Big Crunch [BB-BC] algorithm [10] to increase the performance of the search. Simply the BB-BC method is based on intermediate local searches over the best member of the population. Thus if the individual improves, then this improved one passes to the next generation together with the best solution. In our approach, we adopt this method to the classification case and in addition to the local searches around constants we expand the definition and do multiple local searches among grouped similar elements of an individual. We group elements of an individual as constants, terminals, variables and functions. The search around constants was performed as described in the definition of the BB-BC and we have used gauss distribution where the variance σ=1. For variables and functions we have used point-mutation on variable and function vectors. We execute local search 50 times for constants and 30 times for the others at each of the iterations. Adding such kind of intermediate searches around best-found result enables us to search the space around best solution by concentrating only on a single dimension and without affecting the traditional progress of the genetic evolution.

3 Feature Extraction

Quality of the features is critical for the performance of a classifier, and especially for image classifiers. The quality of a feature refers to that the feature should represent

some distinctive characteristics of the data. The number of features, is related with the phenomena of curse of dimensionality, and the cost of the extraction process are the other two main issues that one should consider while designing an effective feature set. By considering all these issues we built an automatic feature extractor for cell images to find 13 features for each sample. Most of these features are common ones in image processing applications as in [11]. In the remaining of the section first the pre-processing work is defined to make the data ready to extract the features and later the features are explained.

The pollen cells are expected to have texture information since first and second order statistics features are obtained from intensity values. In order to extract the first order statistics of pollen cells, the background of the images must be eliminated and the entire cell region must be determined. Thus, some well-known image processing algorithms are applied to the database to segment the cell region. A sample from the training dataset, *Polypodium Vulgare,* is shown in Fig1. Then an edge detection algorithm is applied and the gaps occurred in the boundaries are filled to find the boundary of the cell. After, the resultant image is dilated and the interior gaps are filled. Although the cell object is almost completely segmented, still there may be some other objects connected to the borders. To remove these effects object images are smoothed. Thus we obtain the texture region to be defined by the features.

Fig. 1. Left: Original cell image Right: Detected cell image

After all pollen cells are detected; the area, the fourth moment and the entropy of the pollen cell are calculated. The area of the cell is the number of the pixels in the detected cell region. The fourth moment gives the information about the sharpness of the histogram of the cell image which is a distinctive property for different cell types. Moreover, uniformity of the histogram of the cell image is obtained by entropy calculation. The last feature is obtained by using Gabor filter as detailed in [11].

However, the first order statistics provide information related to the gray level distribution of the image, but it is not possible to obtain information about the relative positions of the various gray levels in the image. Hence, the features based on second order statistics and defined in co-occurrence matrices are extracted.

The co-occurrence matrix is calculated by using the relative distance among the pixels and their relative orientation. The orientation is denoted by θ and the relative distance is denoted by d. The orientation is quantized in three directions; horizontal, diagonal and vertical. Two dimensional histograms are defined by using d and θ as;

$$P_{\deg(0)}\left(I(m,n)=I_1, I(m\pm d,n)=I_2\right)=\frac{\#\text{ of pairs at distance d with values } (I_1,I_2)}{\text{total }\#\text{ of possible pairs}} \qquad (4)$$

where $P_{deg(0)}(i,j)$ is the (i,j) th element of the co-occurrence matrix for degree 0, $I(m,n)$ is the gray level of the $(m,n)^{th}$ pixel. Similarly;

$$P_{\deg(45)}\left(I(m,n)=I_1, I(m\pm d, n\pm d)=I_2\right)$$
$$P_{\deg(90)}\left(I(m,n)=I_1, I(m, n\pm d)=I_2\right) \quad (5)$$

Features based on co-occurrence matrices are calculated by selecting θ as 0^o, 45^o and 90^o and d as 2 where 0^o is the horizontal, 45^o is the diagonal and 90^o is the vertical directions of the image pixels. Next, basing on this matrix these additional features were extracted for each angle used in co-occurrence matrix.

Contrast (*CON*): The measure of local grey level variations are calculated. For each angle contrast of the co-occurrence matrix is evaluated as;

$$CON = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} P(i,j) \right\} \quad (6)$$

where N_g =256 ($I(m,n) \in \{0,1,2....255\}$)

Inverse Difference Moment (*IDF*): This feature gives high values for low-contrast images. The feature is extracted as;

$$IDF = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{P(i,j)}{1+(i-j)^2} \quad (7)$$

Angular Second Moment (*ASM*): The smoothness of the image can be extracted by using angular second moment feature thus defined as;

$$ASM = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (P(i,j))^2 \quad (8)$$

Whole set of features used in the experiments are listed in Table1 together with their labels;

Table 1. List of features together with their labels

X1	Contrast for degree(0)	**X8**	Inverse difference moment for degree(90)
X2	Inverse difference moment for degree(0)	**X9**	Angular second moment for degree(90)
X3	Angular second moment for degree(0)	**X10**	The area of the pollen cell
X4	Contrast for degree(45)	**X11**	Fourth moment of the pollen cell
X5	Inverse difference moment for degree(45)	**X12**	Entropy of the pollen cell
X6	Angular second moment for degree(45)	**X13**	Gabor filter feature
X7	Contrast for degree(90)		

As the last stage of feature extraction we normalize each feature in their own space into an interval which is common for all features.

4 Experimental Setup

In our experiments we have worked on the cell image database of Bangor/Aberystwyth Pollen Image Database[12]. The database contains huge number

of pollen cell images of 80x100 spatial resolution[1]. Four types of pollen images were used for classification, Figures (3, 4, and 5).

50 samples for each class were used for training purposes and 200 images were extracted for test purposes. Thus, we used totally 250 images from the database for each class. 13 features were extracted for each image. Since GP classifier defines a stochastic process, we employed five independent runs to determine the result of an experiment and the results were obtained by averaging the best of five independent runs. Each run was executed along a fixed number of iterations (generations) with the parameters listed in Table2.

Table 2. GP Parameters of Training Experiments

Population size	80	**Initial Function Number**	20
Crossover rate	0.85	**Depth**	8
Mutation rate	0,25	**Maximum Constant**	5.0
Generation number	1500	**Functions**	Add,Subt,Mul,Div,Max,Min

Fig. 2. Training images from *Quercus robor(QR)* pollen type

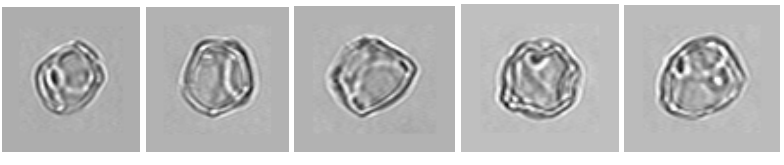


Fig. 3. Training images from *Alnus glutenosa* pollen type

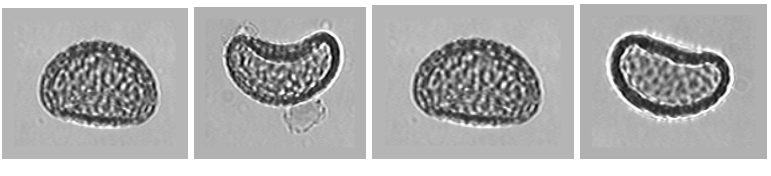


Fig. 4. Training images from *Polypodium vulgare* pollen type

[1] Since the database was built by automatic segmentation of the cell images, there are some outliers that are quite far from representing a cell. We have discarded such samples in our experiments.

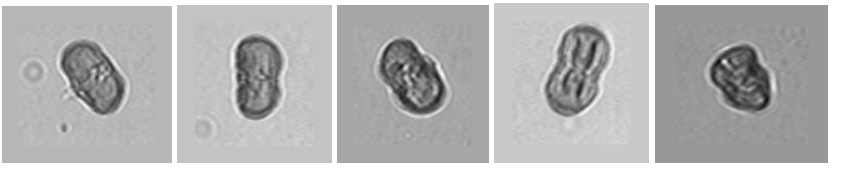
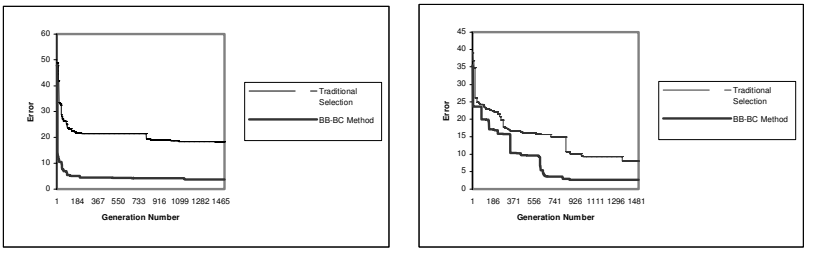


Fig. 5. Training images from *Conopodium majus* pollen type

5 Test Results

In this section, first the result graphics regarding the performance of the designed GP algorithm are presented. Then, the effect of the proposed reproduction scheme is evaluated. Next, the results of the experiments are presented as in the form of confusion matrices. Lastly, the performance is compared with past works.

An efficient evolutionary algorithm is expected to converge to better solutions and never lose good candidates [13]. In Fig. 6 (a) and (b) fitness values of best solutions that were found at each iteration of an execution is figured out for *Quercus robor and Polypodium vulgare* pollen types respectively.



(a) (b)

Fig. 6. Error versus generations for a) *Quercus robor* b) *Polypodium vulgare*

As it can be seen from Fig. 6 (a) and (b) the GP with BB-BC method converges to optimal solution much faster than the traditional selection mechanism.

In Fig. 7, the effects of the proposed reproduction scheme are illustrated. The test results of the two algorithms applied to *Quercus robor* and *Polypodium vulgare* classes are shown in separate rows. Each row consists of two graphics. The left one is consists of the test results of the traditional elitist reproduction method and the right one is obtained from the proposed reproduction scheme.

In Fig.7, the x axis corresponds to each test image and the y axis denotes the labels assigned by the classifier. As aforementioned, to avoid slow convergence, the class labels are determined considering the parameters of the algorithm. In this experiment we consider 0 and 5 values referring 'not-found' and 'found' class labels. As it can be seen from the figures, the new approach improves the convergence of the test results to the true values. However the results of the traditional elitist method may increase false classification probability.

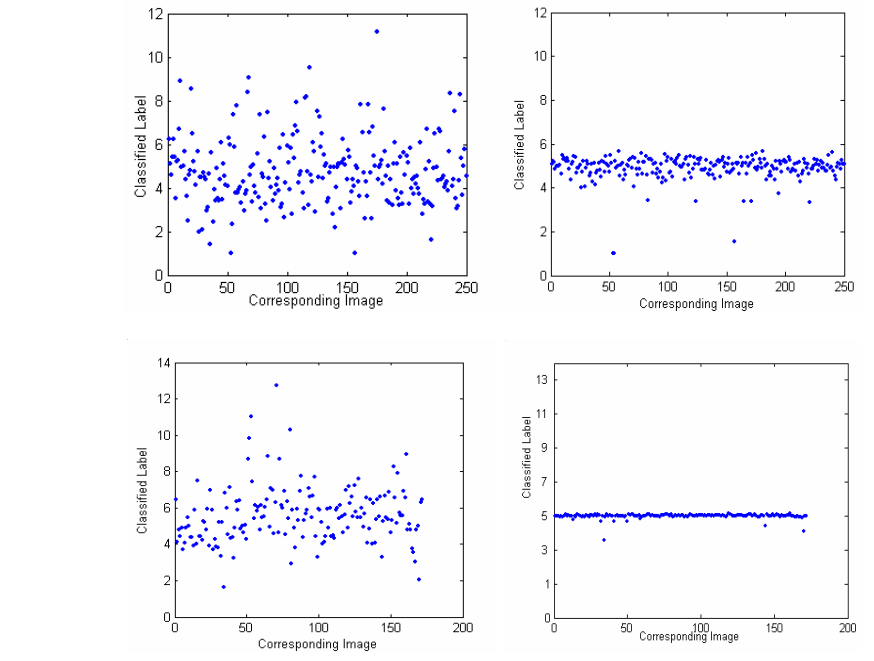


Fig. 7. Effects of the reproduction schemes for classification. (First columns are with elitism and the second ones are with BBBC).

The classification results during the experiments are obtained by averaging the best of the 5 independent runs. The results for each evolved programs for every classes have similar classification performance. In Table3 performance of the training results for each reproduction type is shown.

Table 3. Training performances for conventional and proposed reproduction scheme

Training Performance	Conventional Reproduction	Proposed Reproduction
Quercus robor	92 %	96%
Alnus glutenosa	96 %	96%
Podium vulgare	100 %	100%
Conopodium majus	98 %	100 %

As shown in Table3 the training performance for BB-BC method is minimum 96%. This means that only two samples of the 50 training images are misclassified.

We did binary class experiments, so the results are shown in the form of confusion matrix as in Table4. The confusion matrix contains the results of the test sets of each class type that are applied to the found classifiers of the classes. In the confusion matrix the rows correspond to the true class of the cell and the columns correspond to

the predicted classes. The results for proposed method and conventional method are given in the same matrix. First results in each cell correspond to the proposed method and the others correspond to the conventional method. For example the cell row 4, column 1 means that 0% of the *Conopodium majus* misclassified as *Quercus robor* by using the proposed reproduction method. On the other hand 1.2% of the *Conopodium majus* misclassified as *Quercus robor* by using the conventional reproduction method. Similarly the cell row 3, column 3 means that 99.42% of the *Podium vulgare* correctly classified as *Podium vulgare* by using the proposed reproduction scheme. Table4 shows that the successful classification performance is about 96% on average while the misclassification is nearly 0%.

Table 4. Confusion Matrix of the results for proposed reproduction scheme versus conventional reproduction scheme

Classification Proposed / Conventional	**Quercus robor(%)**	**Alnus glutenosa(%)**	**Podium vulgare(%)**	**Conopodium majus (%)**
Quercus robor	**96.4 / 90.8**	0.4 / 0.4	0.58 / 0	0.096 / 0.4
Alnus glutenosa	0.4 / 1.2	**96 / 99.2**	0 /0.58	0 /0.4
Podium vulgare	0 / 0	0 / 0	**99.42 / 100**	0 / 0
Conopodium majus	0 / 1.2	0 / 0	0 / 0	**99.6 / 97.6**

As illustrated in Table4 the true classification and the false classification performance for the traditional reproduction scheme is nearly same to the proposed method. However the proposed scheme has significant increase in the time of convergence since it allows to use distinct class labels and local search around good candidates.

In Table5 an evolved GP program by using the proposed reproduction scheme for *Quercus robor* image class is represented. The program implies the proposed features with some constants and can accurately classify the image type with small false classification.

Table 5. Evolved GP by using proposed reproduction scheme

```
divide( min( divide( multiply( max( divide( divide(
subtract( divide( divide( subtract( add( multiply( add(
divide( multiply( divide( divide( X3( ), X5( ) ), X6(
) ), X6( ), X3( ), real( 2.962 ) ), X10( ) ), X3( ),
X6( ) ), X10( ), X13( ) ), X10( ), X9( ), X7( ),
X7( ) ), add( X3( ), X13( ) ), X11( ), X7( ), X4( )
), X11( ) ), X9( ) ), real( 11.790 ), X6( ) ), X1( ) ),
X6( ) ), X9( ) ), X6( ), X3( ), X7( ) ), X12( ) ),
X11( ) ), X11( ) )
```

6 Conclusion

In this study, we worked on the problem of automated classification of cell images. First, an effective set of image features are extracted. Although most of the features

are the ones commonly used in literature, the combination of them were considered as an effective feature set definition for such kind of image classification problems. After data pre-processing, then we have designed a GP algorithm for the classification of image samples. In our design, we have addressed some drawbacks of traditional usage of GP in classification problems while suggesting a new fitness calculation scheme and a new reproduction method to overcome these drawbacks. Results showed that the proposed design of the algorithm improves the performance of the cell image classification considerably.

References

1. Koza, J.R.: Genetic programming: On the programming of computers by means of natural selection, Cambridge. MIT Press, London (1992)
2. Zhou, C., Xiao, W., Tirpak, T.M., Nelson, P.C.: Evolving Accurate and Compact Classification Rules ith Gene Expression Programming. IEEE Transactions On Evolutionary Computation 7(6), 519–531 (2003)
3. Bojarczuk, C.C., Lopes, H.S., Freitas, A.A.: Genetic programming for knowledge discovery in chest pain diagnosis. IEEE Engineering in Medicine and Biology Magazine 19(4), 38–44 (2000)
4. Winkeler, J.F., Manjunath, B.S.: Genetic Programming for Object Detection. In: Proc. Genetic Programming Conference, Stanford, CA (1997)
5. Smart, W.R., Zhang, M.: Classification Strategies for Image Classification in Genetic Programming. In: Smart, W.R., Zhang, M. (eds.) Proceeding of Image and Vision Computing Conference, Palmerston North, pp. 402–407 (2003)
6. Kishore, J.K., Patnaik, L.M., Mani, V., Agrawal, V.K.: Application of Genetic Programming for Multicategory Pattern Classification. IEEE Transactions On Evolutionary Computation 4(3), 242–258 (2000)
7. Rodriguez-Dami'an, M., Cernadas, E., S'a-Otero, P.: Pollen Classification Using Brightness-based and Shape-based Descriptors. In: Proceedings of 17th International Conference on Pattern Recognition, Cambridge, vol. 2, pp. 212–215 (2004)
8. France, I., Duller, A.W.G., Duller, G.A.T., Lamb, H.F.: A New Approach to Automated Pollen Analysis. Quaternary Science Reviews, 537–546 (2000)
9. France, I., Duller, A.W.G., Lamb, H.F., Duller, G.A.T.: A comparative study of model based and neural network based approaches to automatic pollen identification. In: France, I., Duller, A.W.G., Lamb, H.F., Duller, G.A.T.A (eds.) British Machine Vision Conference, pp. 340–349 (1997)
10. Eksin, I., Erol, O.K.: A New Optimization Method: Big-Bang Big-Crunch. In: Advences in Engineering Software, Elsevier, Amsterdam (2005)
11. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Academic Press, London (2003)
12. Duller, A.W.G., Duller, G.A.T., France, I., Lamb, H.F.: A pollen image database for evaluation of automated identification systems. Quaternary Newsletter 89, 4–9 (1999)
13. Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D.: Genetic Programming: An Introduction. Morgan Kaufmann, San Francisco (1998)

Use of Radio Frequency Identification for Targeted Advertising: A Collaborative Filtering Approach Using Bayesian Networks

Esma Nur Cinicioglu, Prakash P. Shenoy, and Canan Kocabasoglu

School of Business, University of Kansas, 1300 Sunnyside Avenue,
Summerfield Hall, Lawrence, KS 66045-7585, USA
{esmanur, pshenoy, canan}@ku.edu

Abstract. This article discusses a potential application of radio frequency identification (RFID) and collaborative filtering for targeted advertising in grocery stores. Every day hundreds of items in grocery stores are marked down for promotional purposes. Whether these promotions are effective or not depends primarily on whether the customers are aware of them or not, and secondarily whether the customers are interested in the products or not. Currently, the companies are incapable of influencing the customers' decision-making process while they are shopping. However, the capabilities of RFID technology enable us to transfer the recommendation systems of e-commerce to grocery stores. In our model, using RFID technology, we get real time information about the products placed in the cart during the shopping process. Based on that information we inform the customer about those promotions in which the customer is likely to be interested in. The selection of the product advertised is a dynamic decision making process since it is based on the information of the products placed inside the cart while customer is shopping. Collaborative filtering will be used for the identification of the advertised product and Bayesian networks will be used for the application of collaborative filtering. We are assuming a scenario where all products have RFID tags, and grocery carts are equipped with RFID readers and screens that would display the relevant promotions.

Keywords: RFID, targeted advertising, Bayesian networks, learning Bayesian networks, collaborative filtering.

1 Introduction

Love it or hate it, grocery shopping occupies a significant amount of time of your life. It may seem as a straightforward task—all you need is just a shopping list. However, almost 60% of household supermarket purchases are unplanned and the result of in store decisions [Inman and Winer, 1999]. Even having a shopping list is sometimes not enough. The huge variety of products offered turns the grocery stores into labyrinths, so you have to be cautious not to get lost between the aisles as you search for the products on your list. By the time you find the item you are looking for, you may be overwhelmed to see how many different brands offer the same item.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 889–900, 2007.

Grocery basket selection can be thought as a reflection of customers' needs. Ideally, the products selected should represent the results of a comparison made by the customer based on the price and quality aspects of the products. Considering the nature of a simple grocery-shopping trip described earlier, a careful selection of products requires the devotion of a significant amount of time and energy on the customers' side. On the contrary, modern life imposes time constraints on the customers, which make them unwilling to spend any more time for grocery shopping than is necessary. As a result, the explosion of the size of product assortments (more than 100,000 references in a large hypermarket) no longer allows for a clear identification of differences in quality and prices inside the product mix [Bruno and Pache, 2005].

The situation on the retailers' side is also not very promising. The competition between the grocery stores is increasing every day, forcing the retailers to find new ways to influence the purchase decisions of the customers. Today a huge variety of methods to track and analyze the customers' behavior in e-commerce systems is available. For instance, amazon.com makes real-time recommendations (Customers who bought this item also bought…) to its customers based on the information of the products that have been put in a shopping cart or reviewed by the customer. However, in traditional retail stores, such systems are not used, and, therefore, the customer's behavior is considered as a black box" [Decker, 2003].

As a way to affect the consumers' purchase decisions and to introduce new products, the shelf configurations of the stores are periodically rearranged. Although this might help the retailers to find the optimal allocation of the products, it bothers the customers for not being able to find the products they are looking for.

Another way to influence the customers' purchase decisions is to do promotions. Every day hundreds of items inside the grocery stores are advertised as an effort to trigger the demand of customers for those products on promotion. Whether these special offers will become a subject of interest to customers primarily depends on whether the customers are aware of them or not. Studies suggest that more than half of the shoppers who purchased an item that was on sale were unaware that the price was reduced [Mittal, 1994]. To inform the customers about your ongoing promotions you may increase the rate of your advertisements, which will increase your costs significantly.

Advertisements and promotions are two effective ways to influence sales. However, an advertisement of a promotion will be more successful, if the promotion is particularly advertised to those shoppers who are likely to be interested in the offer. Clearly, the purchase of a product on promotion by an informed customer does not only show the success of the promotion; it is also a valid indicator of customers' interest in that particular product. Thus, the success of a promotion secondarily depends on whether a customer is interested in that product or not.

In order to understand the underlying patterns of customers' purchase decisions, most grocery stores identify its customers through customer loyalty cards via which they keep track of the products purchased by the customer. Based on this information they tailor promotions to individual customers by giving discount coupons at the checkout. However, these promotions happen after the shopping is over which tremendously reduces the impact of the promotion on the sale. The companies are mostly incapable of influencing the customers' decision making process when they

are shopping since the data about the customers' shopping behavior is only available after the decisions are made, i.e., after the shopping is over.

As a way to interact with the customer during the shopping, grocery stores install kiosks from which customers can get information about the ongoing promotions and the products displayed. However, stress and time pressure potentially force a customer to fully concentrate on the original task where the customer is not willing or able to learn the operation of a complex shopping support system [Schneider, 2004].

Having considered all of this, the e-commerce seem to have a huge advantage over traditional grocery shopping because of their capability to make targeted advertising at the same time as the consumer is shopping. Inspired by the real-time recommendation systems of e-commerce we should be looking for ways to transfer the methods of e-commerce systems to the current state of grocery shopping. The capability of RFID technology to identify individual products and collect real time data about the customer behavior inside a store makes a new model for the traditional grocery shopping feasible.

An outline of the remainder of the paper is as follows. In Section 2, we introduce the RFID technology and describe its capabilities in the domain of operations management. In section 3, we discuss 'collaborative filtering' as a way to identify our recommendations based on the customers' preferences and describe our proposed model for the grocery stores. This model promises to enable the grocery stores to make real time targeted advertising. In order to illustrate the working mechanism of our model we are going to use the data set available for the Netflix prize competition. The details of the data set used are described in section 4. Using WinMine toolkit a Bayesian net will be learned from the data set used. As the next step, using the table distributions learned with WinMine, the same BN will be built in Hugin, a commercial software, which allows us to predict the customers' preferences of the movies based on the given information. In section 5, we illustrate the use of our model via different cases. Finally, in section 6, we summarize and conclude.

2 Radio Frequency Identification

Radio Frequency Identification (RFID) is a generic term for a variety of technologies that use radio waves to automatically identify individual items" [Cavoukian, 2004]. This technology known for over 50 years, prepares to have its real bang in the business world after its potential for commercial applications has been realized. The capability of identifying individual products, ability to track the products through the processes, differentiates RFID from its preceding alternatives; but the real and huge potential of RFID systems is hidden in the massive amount of data that is captured by RFID systems.

An RFID system consists of two basic parts: a tag and a reader. Readers, depending upon design and technology used, may be a read-only or a read-write device [Finkenzeller, 1999]. They capture the information stored or gathered by the tag. The RFID tags can be either active or passive, depending whether they have their own power supply or not. Active RFID tags offer superior performance. Because they are connected to their own battery, they can be read at a much higher range-from several kilometers away. However, they are larger and more expensive. Passive tags

have no power source and no on-tag transmitter, which gives them a range of less than 10-meters and makes them sensitive to environmental constraints [Cavoukian, 2004].

Among the automatic identification systems, barcode technology has been the leader for over 20 years. Nevertheless, with the decreasing cost of the RFID tags, companies have begun to favor RFID systems over barcode technology. Although it is a fact that the reduced costs of the RFID tags have contributed a lot to the present popularity of the RFID systems, this is not the main motive why the RFID systems are preferred over barcodes. In Table 1, we illustrate the potential benefits that companies may achieve in their operation management activities by using RFID systems instead of the barcodes.

Table 1. The potential benefits of RFID systems in operations management activities

	Barcode	RFID	Potential benefit of RFIDs
Data capturing capacity	A barcode can hold only around 1000 characters of data. [Mital, 2003]	Up to 128,000 characters in an RFID chip [Mital, 2003].	The superior data capturing capacity of RFID systems offers enough room for a unique serial number, expiration date or other pertinent information [Sweeney, 2005] -This is a serious drawback of bar codes compared to the RFID systems since in RFID systems information is specific to that individual item [Cavoukian, 2004].
Cost	The barcode system is still a much cheaper identification system than the RFID technology and the experts predict that it will remain to be so.	Today Passive RFIDs sell for less than 50 cents in high volumes, and analysts predict they'll sell for five cents in high volumes by the end of this decade [Dipert, 2005].	Tags are reusable and have very long lives, so in supply chain operations where containers are continually reused, there would be no need to re-label the containers, saving on manpower and other costs associated with label production and fixing [Hopwood, 2005].
Processing times	Only one item can be read at a time because of the line of sight technology required. -The existence of dirt or dust can avoid the reading barcodes.	RFID tags can be read in harsh environments such as snow, fog, etc. with a reading distance ranging from 50 feet to 100 meters and beyond [Cavoukian, 2004]	The processing times of items increases significantly, when bar code systems are in use.
The query of components and subassemblies	Requires positioning the cases so that the labels can be read by the scanners -line of sight reading is required	Automatic check that all items from the bill of material -are received -are placed in the right location -RFID does not require positioning the cases	Convenience in order processing -helps to decrease the labor costs -reduce the order preparation times [Rutner, 2004]
A valid source of information in order preparation and processing	not applicable	How much time a worker spends on the preparation of a particular item can be measured	Management could use this data for -setting benchmarks -evaluating employees -planning labor requirements [Rutner, 2004]
Prevention of Spoilage	not applicable	Sensor-equipped tags can monitor the environment surrounding perishable items and maintain a history of environmental changes	RFID systems can be used -to detect potential spoilage conditions [Curtin, 2005] -to identify the causes of spoilage
Prevention of Theft	not applicable	The capability to locate every individual product within the inventory	Provides a tremendous opportunity for companies to prevent theft
Prevention of Shrinkage	Real time data is not available	Automatic collection of real time data	-the automatic collection of real time data prevents the shrinkage problem, and if not, makes the data available to detect the cause of shrinkage -better replenishments decisions can be made since accurate data are readily available with RFID [Lee, 2004]
Prevention of Stockouts	Captures information on how much is sold form each product	Captures information about the real time data of the current inventory (how much is sold, how much is missing)	The ability of RFID systems to prevent and detect when theft and/or shrinkage is present, makes the data more accurate thus preventing the occurrence of stockouts

The use of RFID systems in commercial applications is an emerging trend and RFID is ready to place itself as the dominant technology used in real word applications. However, the importance of RFID technology is not just limited by the convenience it provides. More importantly, RFID systems create massive amounts of data, which gives the ability to track and trace materials at the case-level within the supply chain, and at the item level from manufacturing to post sales. Therefore, the real question to be answered is: How can we transform this massive amount of data into managerially useful information?

3 Collaborative Filtering

As mentioned earlier the real potential of RFID systems is hidden in the massive amount of data collected through RFID. This application is a perfect illustration of that. We can use the RFID technology for getting real time information about the consumer behavior as they are shopping and that may enable us to inform the customer about the promotions in store in which the customer is likely to be interested. Using RFID we can get information about the products a customer is placing in his shopping basket, and using collaborative filtering we can advertise those products on promotion which the customer is more likely to be interested based on what is already in the customer's shopping basket.

Collaborative filtering, first introduced by Resnick *et al.* (1994), is defined as predicting preferences of an active user given a database of preferences of other users [Mild, 2002]. Depending on the technology used, recommendation systems are classified in two classes, content-based filtering (CBF) and collaborative filtering (CF). Content-based methods make recommendations by analyzing the description of the items that have been rated by the user and the description of items to be recommended [Pazzani, 1999]. The main difference between collaborative filtering and content-based filtering is that CF does not rely on the content descriptions of the items, but depends purely on preferences expressed by a set of users [Yu *et al*, 2004]. Since collaborative filtering does not depend on error-prone machine analysis of content, it has significant advantages over traditional content-based filtering (ability to filter any type of content, etc.) [Herlocker *et al.*, 2000].

Popescul *et al*. [2001] describe a unified collaborative and content-based system. de Campos *et al*. [2006] describe a Bayesian network model for hybrid collaborative and content-based filtering. Adomavicius and Tuzhilin [2005] survey the field of collaborative, content-based, and hybrid recommender systems. Zhang and Callan [2001] developed an algorithm for setting dissemination thresholds while filtering documents. Linden *et al.* [2003] compare traditional collaborative filtering, cluster models and search based models with their method item-item collaborative filtering.

In e-commerce, collaborative filtering is widely used as a tool for targeted advertising. Using the capabilities of RFID, we might be able to transfer this method to traditional retail stores and base the advertisements on real-time data.

The technique used in collaborative filtering is based either on explicit or implicit voting. The data sets in explicit voting contain users explicit preference ratings for products. Implicit voting refers to interpreting user behavior or selections to impute a vote or preference [Breese *et al.,* 1998]. Our case is an example of implicit voting,

since our model will use binary choice data that identifies whether a product is placed in cart or not.

The model we are proposing is as follows. All products in a grocery store are equipped with RFID tags. The carts in the grocery store are equipped with RFID scanners, which are utilized to collect information about the products that are placed in a customers' cart. In each cart, there is also a screen where the promotions are displayed. The basic idea of our model is to inform the customer about those products on promotion that the customer is likely to be interested in buying based on the products already in the cart. The selection of the product advertised is a dynamic decision making process since it is based on the information of the products placed inside the cart while customer is shopping. Collaborative filtering will be used for the identification of the advertised product and Bayesian networks will be used for the application of collaborative filtering.

At the beginning of the shopping process, there are no products in the cart. At this stage, the system can just display those products on promotion that have the highest marginal probabilities. As the customer places products in the cart, the system can display those products that the customer is likely to be interested in purchasing based on items in the cart.

4 Dataset

The proposed model above requires data captured through RFID systems for the different market baskets of the customers. Since we did not have access to a grocery basket dataset, we decided to use the publicly available Netflix prize competition[1] dataset to illustrate our application [Netflix, 2007]. While it is not quite the same, we were able to convert the Netflix dataset to a basket dataset with movies as the products instead of grocery items.

The training data set of the Netflix prize competition constitutes of 17,770 files, one per movie. Each file contains customer ID, the rating given by the customer, and the date of the rating. The ratings are on a scale from 1 to 5, 5 as being the best rating possible. For the analysis done in this paper, 1,695 movie files from this training data set have been chosen on a random basis. These separate data files are merged into a big data set where the ratings for the movies are sorted based on the customer ID and the date has been dropped out.

The goal of our model is to predict the products that the customer may be interested in based on the products that (s)he has placed in the cart. Trying to interpret the customers' behavior suggests the need for implicit voting instead of a detailed 1 to 5 rating scale. Hence, we transformed our data set into a new data set where the ratings 3, 4 and 5 are replaced by 1's as an indicator of the movies being in a user's basket. If the customer has rated the movie as 1 or 2 or has not rated the movie at all, then the movie rating is replaced with a zero, which means that it is not in the cart. Here we are assuming that the movies not rated by the customer are movies that are not in the customer's cart.

[1] The Netflix prize competition seeks to substantially improve the accuracy of predictions about how much someone is going to love a movie based on the ratings of the movies they have already seen.

In a grocery store, there are literally hundreds of thousands of different products. For the problem of finding associations between the products that are in carts, we need to aggregate the products. For example, tomato sauce may be sold in different brands, different sizes, different packaging, etc., and all of these need to be aggregated into a single product. [2] The problem of finding a good aggregation can be a difficult one. We need to decide on a number of aggregated products, and a technique to do the aggregation. Cluster analysis from multivariate statistics is one method that can be used for doing the aggregation. The optimal number of aggregated products is an empirical question, and an approximate number can be found by experimentation.

After transformation of the data to the desired format, the next step was to select the movies that are going to be used for creating a Bayes net. In order to select the movies from different groupings we used cluster analysis. The FASTCLUS procedure in SAS was used for cluster analysis, where we limited the maximum number of clusters obtained to thirty[3]. As a result, we obtained thirty different clusters and chose one movie from each cluster on a random basis. The final data set used to build the Bayes Net constitutes of the movie preferences of 65,535 users for the thirty movies selected. The set of movies selected appears in the Bayes net model shown in Figure 1 below.

Our motivation for learning a Bayes Net is to find the predictive relationships between the movies based on the movies liked or disliked by the customer. WinMine [Heckerman *et al.*, 2000], a tool developed at Microsoft Research, is used to learn a Bayes Net. Using WinMine, the data is divided into a training set and a test set. We performed a 70/30 train/test split and had 45,874 training cases and 19,661 test cases. All of the variables are used as input-output variables (both predicted and used to predict). To set the granularity of the Bayesian network learnt by WinMine, a factor called *kappa* is used, which is a number between 0 and 1. As *kappa* approaches 1, the model becomes very dense. Since our model is already quite dense, we decreased the value of kappa from its default value of 0.01 to 0.00001. The resulting BN is given in Figure 1.

The accuracy of the learned model on the test set is evaluated using the log score $\text{Score}(x_1, \ldots, x_N) = \frac{\sum_{i=1}^{N} \log_2 p(x_i \mid \text{model})}{nN}$, where n is the number of variables in X, and N is the number of cases in the test set. Our model results in a log score of -0.4169, meaning on average, the log probability that each variable assigns to the given value in the test case, given the values of all other variables, is -0.4169, which translates to a probability of 0.75. Using WinMine we can also compare the difference between the provided model and the marginal model. A positive difference is desired between the provided model and the marginal model, signifying that the model outperforms the marginal model on the test set. In the same way that a regression model

[2] In the first iteration, we selected 33 movies that had a large number of user ratings (without doing cluster analysis) and used it to learn a Bayes net. However, that was not very effective in predicting the baskets of users in the test set (lift over marginal was about 0.04364).

[3] We did not attempt to determine an optimal (or an approximate) number here. We picked thirty for convenience. Since we obtained good results, we did not experiment with other numbers.

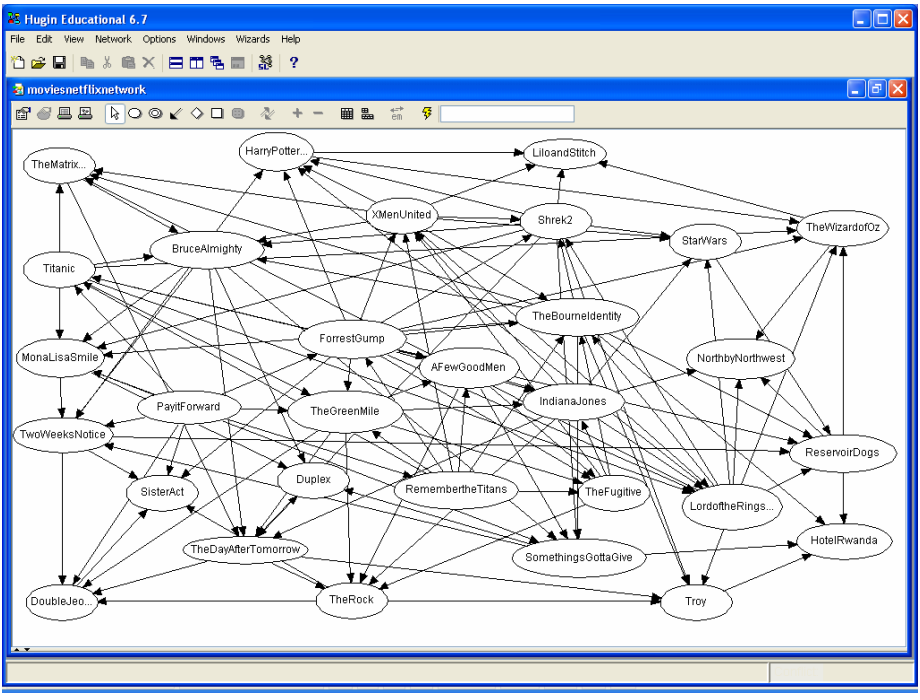


Fig. 1. A Bayes net for 30 movies from the Netflix prize dataset

is more accurate than a simple baseline model chosen in the form of a mean dependent value, the "lift over marginal" log score provides information on how well the model fits the data. The lift over marginal log score in our model is 0.1302, which suggests the performance of our model is quite good. If we ignored the products in the cart and used the marginals for prediction, the average probability of the correct prediction is 0.68 (or log score of –0.5471). Using the products in the cart, the average probability of correct prediction improves to 0.75 (or log score of –0.4169) resulting in a lift over marginal log score of (–0.4169)–(–0.5471) = 0.1302. There are many ways of evaluating collaborative filtering recommender systems (Herlocker *et al*, 2004), and the lift over marginal is a good conservative measure of effectiveness for our application.

5 A Case Study

In the previous section we have illustrated how a BN can be learned using the WinMine toolkit. Using the probability tables constructed by WinMine, we constructed the same Bayes Net in Hugin, a commercial software. The conditional probability table used for the movie 'Lord of the Rings: The Two Towers' is illustrated in Table 2 below.

Table 2. The conditional probability table for *Lord of the Rings: The Two Towers*

Lord of the Rings: The Two Towers																
Forrest Gump	0								1							
Titanic	0				1				0				1			
X-Men United	0		1		0		1		0		1		0		1	
Indiana Jones	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0.88	0.76	0.53	0.21	0.83	0.59	0.37	0.16	0.80	0.52	0.33	0.13	0.70	0.41	0.27	0.10
1	0.12	0.24	0.47	0.79	0.17	0.41	0.63	0.84	0.20	0.48	0.67	0.87	0.30	0.59	0.73	0.90

The advantage of using Hugin is that we are able to enter evidence to the BN and update all probabilities accordingly using the 'sum normal' propagation method. In addition to that, the 'max normal' propagation method allows us to find states to the most probable configuration. The state of node with the most probable configuration is given the value of 100. The values for all other states are the relative values of the probability of the most probable configuration in comparison to the most probable configuration.

By using the sum-propagate normal propagation method without entering any evidence, we obtain the marginal probabilities for all the movies in the BN. The results suggest that for the state '1' the movie 'The Green Mile' has the highest marginal probability 40.57%, and 'Duplex' has the lowest marginal probability 5.23%.

Suppose we want to predict whether a specific customer is going to like the movie *Forrest Gump* or not. Without having any information about the customers' previous movie preferences the marginal probability for the state '1' is 40.43% and the state for the most probable configuration is '0'. Suppose we get the information that the customer rented the movie *A Few Good Men* and liked it. Accordingly, the posterior marginal for *Forrest Gump* increases to 69.75%, the most likely state is still '0'. Next, suppose we get the information that the customer also liked *The Wizard of Oz*. The posterior marginal probability for *Forrest Gump* increases to 90.13% and the most likely state changes to '1'. The results for this case are summarized in Table 3 below.

As our second case, consider a scenario where we need to choose between the two movies *Mona Lisa Smile* and *Lord of the Rings*: *The Two Towers* to recommend to the customer. The initial most likely state is '0' for both movies. Based on their marginal probabilities, which are given in Table 4 below, *Lord of the Rings: The Two Towers* should be chosen for recommendation, since it has a much higher marginal probability for the state '1'.

Suppose we receive information about movie preferences of the customer to whom we are going to make the recommendation. Learning that the customer liked *Pay It Forward*, *Something's Gotta Give*, *Two Weeks Notice* and *Titanic* with the particular order given, changes the posterior marginal probabilities. Until we obtain the information that the customer liked *Something's Gotta Give*, the marginal probabilities indicate that *Lord of the Rings* should be chosen for recommendation. After subsequent observations, *Mona Lisa Smile* takes the lead for recommendation. At the point where we learn that the customer liked *Two Weeks Notice*, the most likely state for *Mona Lisa Smile* becomes '1' where for Lord of the Rings it is '0' still. After we get the information that the customer also liked the movie *Titanic* the most likely state for both of the movies becomes '1'. The details of posterior marginal probabilities and the most likely states are given in Table 4 above.

Table 3. Posterior probabilities and most likely state for *Forrest Gump*

Information & Rating	*Marginal*	*Most likely state*
Prior	40.43%	0
A Few Good Men = 1	69.75%	0
Wizard of Oz =1	90.13%	1

Table 4. Posterior probabilities and most likely states for *Mona Lisa Smile* and *Lord of the Rings: The Two Towers*

	Mona Lisa Smile		*Lord of the Rings: The Two Towers*	
Information & Rating	Marginal	Most likely state	Marginal	Most likely state
Prior	19.28%	0	33.85%	0
Pay It Forward = 1	36.86%	0	48.08%	0
Something's Gotta Give = 1	63.05%	0	61.37%	0
Two Weeks Notice = 1	72.22%	1	63.23%	0
Titanic = 1	76.00%	1	67.38%	1

6 Conclusions and Summary

We have proposed a system using RFID and collaborative filtering for targeting advertising in grocery stores. We have illustrated the use of such a system using the Netflix prize competition dataset.

The proposed model promises to influence the customers' decision-making process while shopping, which will increase the success of the promotions. Also, it is very important to notice that the contribution we will get through the proposed model is not just limited by the improvement of promotions. Transferring the methods of e-commerce to actual retail stores through real time data collection with RFID may give us insight about the operational problems such as the optimal placing of products inside a store. Also, many grocery stores have data on users using loyalty cards. The longitudinal information about these users can be used to further improve the effectiveness of our system. For the next stage of this research, the proposed model will be constructed using real grocery data.

Acknowledgments. We are grateful to John Charnes, Sanjay Mishra, James Lemieux, and two anonymous ECSQARU-07 reviewers for comments and suggestions for improvements.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. on Knowledge and Data Engineering 17(6), 734–749 (2005)

2. Breese, J.S., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: Proceedings of Fourteenth conference on Uncertainty in Artificial Intelligence (1998)
3. Bruno, D., Paché, G.: From Traditional Retailing to E-tailing: The death and Rebirth of the Hypermarket Format? Frontiers of E-Business Research (2005)
4. Cavoukian, A.: Tag, You're It: Privacy Implications of Radio Frequency Identification (RFID) Technology. Information and Privacy Commissioner (2004)
5. Curtin, J., Kaufmann, R.J., Riggins, F.J.: Making the Most Out of RFID Technology: A Research Agenda for the Study of Adaptation, Usage and Impact of RFID. Information Technology and Management (2007)
6. de Campos, L.M., Fernandez-Luna, J.M., Huete, J.F.: A Bayesian Network approach to Hybrid Recommending Systems. In: Proc. 11th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU-06), pp. 2158–2165 (2006)
7. Decker, C., Kubach, U., Beigl, M.: Revealing the Retail Black Box by Interaction Sensing. In: Proceedings of ICDCS 2003, Providence, Rhode Island, pp. 328–333 (2003)
8. Dipert, B.: Reading between the Lines. Business & Company Resource Center (2005)
9. Finkenzeller, K.: RFID Handbook Radio-Frequency Identification and Applications. John Wiley & Son, New York (1999)
10. Heckerman, D., Chickering, D.M., Meek, C., Rounthwaite, R., Kadie, C.: Dependency Networks for Inference, Collaborative Filtering, and Data Visualization. Journal of Machine Learning Research 1, 49–75 (2000)
11. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining Collaborative Filtering Recommendations. In: Proceedings of the ACM 2000 Conference on Computer Supported Cooperative Work, pp. 241–250 (2000)
12. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. on Information Systems 22(1), 5–53 (2004)
13. Hopwood, J.: What is RFID? Intellident, http://www.intellident.co.uk
14. Inman, J., Winer, R.: Where the Rubber Meets the Road: A Model for In-store Consumer Decision Making. Marketing Science Institute Report, 98–122 (1998)
15. Lee, Y.M., Cheng, F., Leung, Y.T.: Exploring the Impact of RFID on Supply Chain Dynamics. In: Proceedings of the 2004 Winter Simulation Conference (2004)
16. Linden, G., Smith, B., York, J.: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing, 76–80 (2003)
17. Mild, A.: An Improved Collaborative Filtering Approach for Predicting Cross-category Purchases based on Binary Market Basket Data. Adaptive Information Systems and Modelling in Economics and Management Science (2002)
18. Mital, T.: The Emergence of RFID Technology. ISCR Future Technology Topic Brief (2003)
19. Mittal, B.: An Integrated Framework for Relating Diverse Consumer Characteristics to Supermarket Coupon Redemption. Journal of Marketing Research 31(11), 533–544 (1994)
20. Netflix (2007), http://www.netflixprize.com/
21. Nogués, M.T.M.: Combining Machine Learning and Rule-Based Approaches in Spanish Syntatic Generation. PhD Dissertation, Institut Universitari de Lingüística Aplicada Universitat Pompeu Fabra (2006)
22. Pazzani, M.J.: A Framework for Collaborative, Content-Based and Demographic Filtering. Artificial Intelligence Review (1999)

23. Popescul, A., Ungar, L.H., Pennock, D.M., Lawrence, S.: Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments. In: Proceedings of 17th Conf. on Uncertainty in AI (UAI-01) (2001)
24. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: An Open Architecture for Collaborative Filtering of Netnews. In: Proceedings of the ACM Conference on Computer Supported Cooperative Work, pp. 175–186 (1994)
25. Rutner, S., Waller, M.A., Mentzer, J.T.: A Practical Look at RFID. Supply Chain Management Review (2004)
26. Schneider, M.: Towards a Transparent Proactive User Interface for a Shopping Assistant. In: Workshop on Multi-user and Ubiquitous User Interfaces, pp. 31–35 (2004)
27. Sweeney II, P.J.: RFID for Dummies. Wiley, Chichester (2005)
28. Yu, K., Schwaighofer, A., Tresp, V., Xiaowei, X., Kriegel, H.P.: Probabilistic Memory-based Collaborative Filtering. IEEE Transaction on Knowledge and Data Engineering 15(1), 56–69 (2004)
29. Zhang, Y., Callan, J.: Maximum Likelihood Estimation for Filtering Thresholds. In: Proc 24th Ann. Int'l ACM SIGIR Conf., pp. 294–302. ACM Press, New York (2001)

Development of an Intelligent Assessment System for Solo Taxonomies Using Fuzzy Logic

John Vrettaros[1,2], G.Vouros[2], and A.S. Drigas[1]

[1] NCSR DEMOKRITOS, Department of Applied Technologies,
Patriarhou Grigoriou, 15310 Ag. Paraskevi , Greece
[2] Aegean University, Info and Communication Systems Eng,
83200, Karlovassi, Samos, Greece
{dr,jvr,imm}demokritos.gr, georgev@aegean.gr

Abstract. In this paper is presented a modeling of assessment systems of taxonomies using fuzzy logic. Specifically the taxonomies system solo is studied, which can be applied in a wide range of fields of diagnostic science. In what concerns education, the test correction is extremely hard and demands experts that are not always available. The intelligent system offers the opportunity to evaluate and classify students' performance according to the structure of the observed learning outcome, concerning the cognitive development of the students in the field of mathematics. The system was tested on high school and university students.

Keywords: fuzzy logic, intelligent system, neural network, assessment system.

1 Introduction: Intelligent Systems and Problems of Knowledge Assessment

An intelligent system is based on an extended quantity of knowledge related to a certain field of problems. This knowledge is organised as a set of rules that allows the system to inference based on the available data. This knowledge –based methodology used in problem solving and more generally in system design has been an evolutionary change in Artificial Intelligence. The consequences of this change are very important since the traditional form of a program (data + algorithm = program) has been replaced by a new architecture, which has as its core a knowledge base and an inference engine under the form:

Knowledge + Inference = System

The specific problem that we have to solve is the construction of an intelligent system, which will be able to evaluate and classify student according to some features, which will be extracted from their answers, into different levels of knowledge. The results are based on a research carried out on high school students and related to the wider field of Mathematics. The classification problem of educated people in different knowledge levels, the study of the transition between these levels

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 901–911, 2007.

as well as the notional change that takes place when a student stops using a naïve (wrong) model and starts using a scientific (right) model, are three of the most important problems in Cognitive Science. A great number of researchers have proposed different methodologies for knowledge acquisition in different scientific fields (Maths, Physics, etc) based on computational and Artificial Intelligence models[8]. Artificial Intelligence methodologies present great interest in theoretic level since they can deal effectively with complexity and fuzziness, which are two of the most important problems in system theory, strongly bound to reality.

In this specific application, analysis starts with the processing of the answers to carefully selected and formed questionnaires which are filled by students. Certain features are extracted out of this analysis that lead to the classification into levels of five different theme sections: Arithmetic, Algebra, Applications Space Perception, and Probabilities and Data. Next, based on this analysis and rule-based knowledge the student classification takes place. Basically the problems that needs to be solved is the automatic classification of students in different levels, using fuzzy logic and artificial neural nets techniques and aiming at creating a system that unifies symbolic and arithmetic processing. For further research, we could note the use experts' knowledge in order to improve the knowledge of educated people (which means transition to a higher level), study the dynamic evolution of the population of educated people and model the changes that take place. Based on the fact that the problem to be solved is a assessment problem, for which there is no specific theory and its data enclose uncertainty (the problem is not purely computational, there is no mathematical solution and the data are not completely known), we can say that use of an intelligent system is appropriate and leads to the construction of a useful tool for student classification in different levels.

The questionnaires that are filled up by students include the aforementioned five theme sections. Each theme section includes four questions, each one corresponding to one of the following levels of knowledge: Single-Structural (SS), Multi-Structural (MS), Relational (R) and Abstractive(S). It should be noted here that the question that corresponds to the Abstractive level cannot be answered by students of the certain age, and consequently we can say that each theme section has three questions. In addition, if a student does not answer any of the three questions in a theme section, he/she is classified in the Pre-Structural (PS) level.

2 Description of the SOM Algorithm Grading System Modelling

The aim of the automatic grading system is the simulation of the teacher's grading system. The answers of the students to the five theme sections are decided into two different categories: Controversial Answers (CA) and Non –Controversial (NCA). Non-Controversial answers are the answers we can be based on in order to classify the student in a level without any uncertainty [1]. For example, if a student gave the following answers to the section that corresponds to Algebra:

Q4. Wrong, Q5. Wrong, Q6. Wrong

Then the student is classified into the Pre-Structural level in Algebra without any controversy. It his/her answers are:

Q4. Right, Q5. Right, Q6. Wrong

Then the student is classifies into the Multi-Structural level in Algebra, without any controversy again. However, there are some answers base on which we cannot conclude to an automatic classification, and we have to take under consideration other factors (in the same way a teacher acts when grading a students answers). For example if a student answers:

Q4. Wrong, Q5. Right, Q6. Wrong

Then his/her classification into a level of knowledge is not straightforward as it was on the examples mention above.

The automated grading system that was developed is illustrated in Fig. 1. The neural network specifies the level of each student in each theme section in cases of Non -Controversial answers. In cases on Controversial answers we have developed two fuzzy systems, since the classification is not obvious and trying to simulate the teacher's way of grading , taking under consideration numerous factors. This way, we take advantage of the symbolic knowledge of system experts and more specific the rule-based knowledge [2]. The first of the two fuzzy systems is implemented based on some statistical analysis and the analysis of some factors such as the Rigour according to which the grading of the certain answer will be done. The second fuzzy system extracts the level of knowledge at each theme section taking into account the Rigour (which the previous system's output) and the answers given to the questions of the specific theme section. Next, the final level is determined for each student based on the results (outputs) of the above systems.

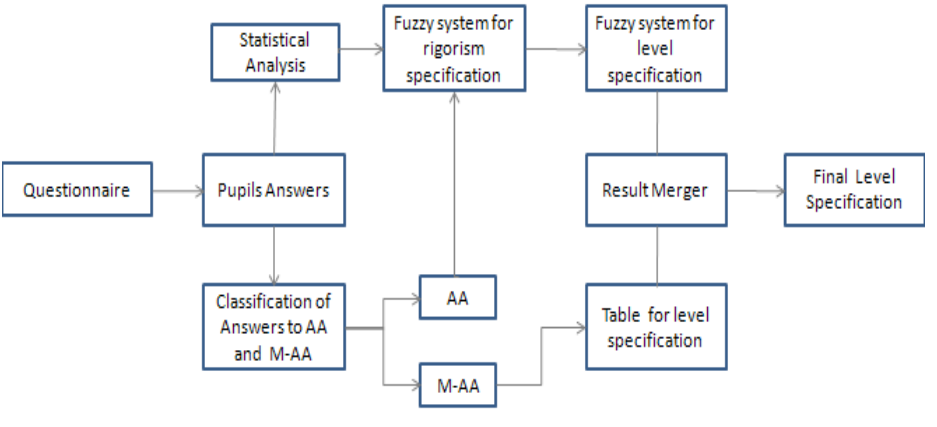


Fig. 1. Grading Modelling System

3 Student Level Determination System for Each Theme Section: Non-Controversial Answers

In the previous section we briefly described the procedure that was followed in order to implement the automated classification of students into levels of knowledge. We mentioned that the students' answers, in the five different theme sections, are divided into two categories: Controversial and Non-Controversial [3]. For the first category, Table 1. was used in order to extract the results. In the specific application, an algorithmic method could also be used for the extraction of the final result. However, based on the fact that we are interested in the extension and application of the developed system in more complicated problems (i.e. we will ask the system to grade the answer using a grade between 1 and 10 or 100), the use of the table is the most appropriate. The Non-Controversial answers are illustrated in Table 1. We have used 0 to symbolize the wrong answer, 1 for the right answer and 2 for invalid answer (case where the student does not answer).

Table 1. Classification based on Non - Controversial answers

SYMBOL	ANSWERS	KNOWLEDGE LEVEL
000	WRONG – WRONG – WRONG	Pre-Structural
100	RIGHT – WRONG – WRONG	Single-Structural
110	RIGHT –RIGHT - WRONG	Multi-Structural
111	RIGHT – RIGHT - RIGHT	Relational
222	INVALID – INVALID - INVALID	Pre-Structural
220	INVALID – INVALID - WRONG	Pre-Structural
202	INVALID – WRONG - INVALID	Pre-Structural
200	INVALID – WRONG - WRONG	Pre-Structural
122	RIGHT – INVALID - INVALID	Single-Structural
120	RIGHT – INVALID - WRONG	Single-Structural
112	RIGHT – RIGHT - INVALID	Multi-Structural
102	RIGHT – WRONG - INVALID	Single-Structural
022	WRONG – INVALID - INVALID	Pre-Structural
002	WRONG – WRONG - INVALID	Pre-Structural
020	WRONG – INVALID - WRONG	Pre-Structural

4 Student Level Determination System for Each Theme Section: Controversial Answers

In the previous section we referred to the cases where the classification of the students into knowledge levels is done based on their answer without any uncertainty. In this section we will refer to the Controversial cases where the student classification in some level cannot be done without any uncertainty [5]. For the evaluation of these answers we will consider the following factors (that correspond to the factor that the teachers take into account when dealing with controversial cases):

1. The difficulty of the certain subject that obviously affects its grading.
2. The number of void answers, which is the number of question that were left unanswered by the student. This factor is considered since it affects the student's evaluation. If, for example, we want to grade a controversial answer (e,g case Q4. WRONG, Q5. RIGHT, Q6. RIGHT) and the student has a great number of unanswered questions, this means that probably the student is not answering the questions randomly, but he/she answers the question seriously. We conclude that probably the wrong answer in Q.4 is wrong due to carelessness, since the right answers in Q.5 and Q.6 (which are obviously much harder to answer that Q.4) are not given by chance. Consequently the student can be classified in the Relational level in the specific theme section..
3. Child level, meaning the general impression the student makes.

Controversial cases occur when a right answer follows a wrong one. These cases are 12 in total, as it is presented in the following table (Table 2):

Table 2. Controversial answers description

SYMBOL	ANSWERS	KNOWLEDGE LEVEL
001	WRONG – WRONG - RIGHT	?
010	WRONG – RIGHT - WRONG	?
011	WRONG – RIGHT - RIGHT	?
101	RIGHT – WRONG - RIGHT	?
221	INVALID – INVALID - RIGHT	?
212	INVALID – RIGHT - INVALID	?
211	INVALID – RIGHT - RIGHT	?
210	INVALID – RIGHT - WRONG	?
121	RIGHT – INVALID - RIGHT	?
021	WRONG – INVALID - RIGHT	?
012	WRONG – RIGHT - INVALID	?
201	INVALID – WRONG - RIGHT	?

In general, we can say that the selection of the level in cases of the controversial answers is different depending on the student. It is affected by the student's answers to previous questions, the number of questions that are left unanswered and the level of the question. In order to model the controversial cases there have been designed and implemented two fuzzy systems, that are analytically described in the next sections [6].

4.1 Rigor Grading Determination Subsystem

The first system evaluates the Rigor according to which the student will be graded. Rigor is a number between 0 and 1 and it is used for the classification of student in knowledge levels. The system has three inputs and one output. The inputs are factors that affect the grading of each controversial answer: number of unanswered question, question level and child level [8]. The output is one: the Rigor. The question level is evaluated according to the answers of other students to this question. The more the

students that answered this question, the easier the question is and the Rigor level is increased. The child level is estimated based on the student's answers.

In this specific case the values of each input are between two values. The difficulty of the subject and the number of void answers take values between 0 and 100, and the Child level take values between 0 and 3. The Difficulty of the subject is estimated based on other students' answers. The x axis –s normalized and has values in the range [0,100]. We can define a partition on the domain field of Subject Difficulty, for example we can say that if value of Difficulty is in [0, 30) then Difficulty is small, if it is in [30, 65), then is said to be medium and finally if it belongs in [65,100), is said to be larger. However this way of classical partitioning introduces great uncertainty in some areas (e.g. close to 30, 65 and 100). This means that if the value of Difficulty is equal to 29 then difficulty is small, whereas if it is equal to it is medium. In order to avoid such problems we define the fuzzy partitions (one for each input) on the domain field of each input A_1, A_2 and A_3. Each fuzzy partition is of order 3. A Fuzzy partition B, of order 3, is defined on the domain field of the output, which is [0, 1] as mentioned above. The fuzzy partitions A_1, A_2 A_3 and B linguistic representations of the domain fields and consequently their elements are linguistic terms of the form SMALL, LARGE, MEDIUM, etc. In Fig. 2. the fuzzy partition of the first input is described.

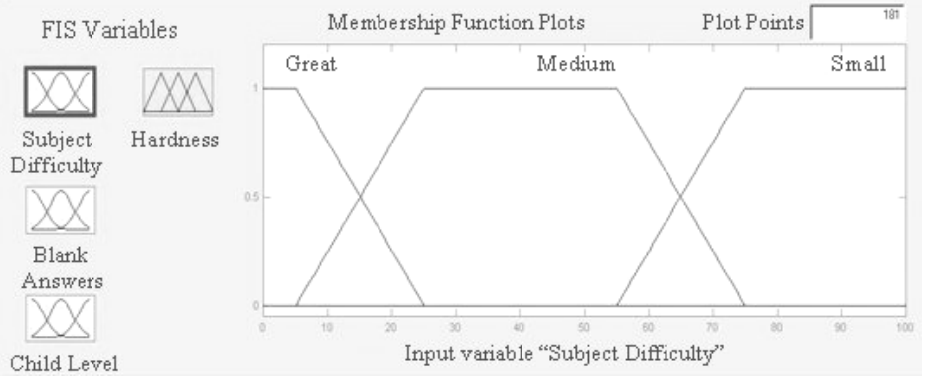


Fig. 2. Fuzzy partition of 1st input: "Subject Difficulty"

If the value of the input is for example equal to 2 or 90, then we can say that the Subject Difficulty is 100% SMALL or 100% LARGE respectively. But if the input value equals 15 then the Subject Difficulty is 0.5 SMALL and 0.5 MEDIUM, and if it is equal to 60 is 0.75 MEDIUM and 0.25 LARGE. The above consideration, that is the use of fuzzy partitions, is obviously much closer to reality since it simulates better the human way of thinking.

The next step is to define the rules of the system. These rules are the following:

1. If Subject Difficulty is "BIG" then Rigor is "HIGH".
2. If Subject Difficulty is "MEDIUM" then Rigor is "ENOUGH".
3. If Subject Difficulty is "SMALL" then Rigor is "LOW"

4. If Void Answers are “FEW” then Rigor is “SMALL”.
5. If Void Answers are “ENOUGH” then Rigor is “ENOUGH”.
6. If Void Answers are “MANY” ” then Rigor is “HIGH”.
7. If Child Level is “LOW” then Rigor is “LOW”.
8. If Child Level is “AVERAGE” ” then Rigor is “ENOUGH”.
9. If Child Level is “HIGH” then Rigor is “HIGH”.

The system’s domain field is shown in Fig. 3.

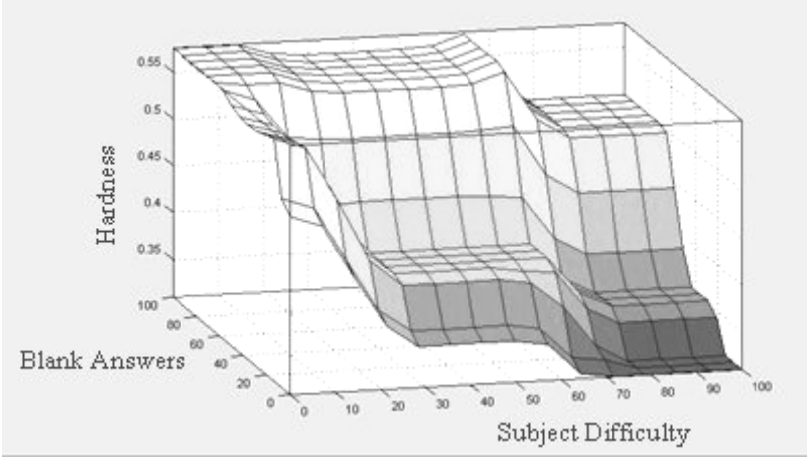


Fig. 3. System’s Domain Field

4.2 Student Level Determination Subsystem Per Theme Section

The second fuzzy system determines the student’s level in each theme section, and more precisely based on the Rigor of grading (that is the output of the previous system) a fuzzy system has been designed student classification in theme sections. In the second fuzzy system, we use only the data of the student under consideration. The two fuzzy systems together with the neural network provide us with the answers of the first part, meaning the student classification in one of the aforementioned knowledge levels per theme section [10].

The system has two inputs and one output. The first input is the Rigor and the second one is the three answers for the subject. The output is a number between o and 3 that corresponds to the four levels of knowledge (Pre-Structural, Single-Structural, Multi-Structural and Relational) for each theme section. The result in controversial cases can be a decimal number.

The values that Rigor is allowed to take are 0.2 τοι 0.65. The values of X-axis are covered by two membership functions. If, for example, we take as input the value 0.3, the Rigor is 50% Low and 50% enough. The second input is the answers of the student under investigation. The input’s values are normalized from 1 to 10. In order

to interpret the student answers in values between 0 and 10 we apply the following formula:

$$\text{Results} = \alpha + \beta + \gamma + \delta$$

where α is the number of right answers, β is the number of last right answer, γ is the number of first right answer δ is the number of void answers. The controversial answers can only have two values otherwise they are not controversial. ($a \leq 2$). The greatest number we can have is 10 and the least 4.

The second input of the system is the student level in the theme section under consideration. The output of the two fuzzy systems, in combination with the output of the neural network, determines the levels of the students in the five theme sections.

The output values are between 0 and 3.) corresponds to Pre-Structural level, 1 to Single-Structural, 2 to Multi-Structural and 3 to Relational. The rules that associate the inputs with the outputs are the following:

1. If Rigor is HIGH then the Level is LOW.
2. If Rigor is AVERAGE then the Level is MEDIUM.
3. If Rigor is LOW then the Level is HIGH.
4. If the Answers are FEW then the Level is LOW.
5. If the Answers are ENOUGH then the Level is MEDIUM.
6. If the Answers are MANY then the Level is HIGH.

5 Final Level Determination

Up to now, we have determined the levels of knowledge of students in five different theme sections. Based on these levels we will determine the final overall level. The final level is a number between 0 and 3 that corresponds to one of the four knowledge levels. In the previous sections we described the procedure of level determination based on the theme sections. The procedure that follows next investigates the students' answers according to knowledge levels rather than theme sections [11]. The degrees of trust will specify at what point the student under investigation belongs to each level. The degree of trust is number between 0 and 1.

The system was divided into four parts, each one associating the number of given answers to the number that we believe it belongs to the specific level (Fig. 4.).

The degrees of trust are three: one for the Single-Structural, one for the Multi-Structural and one for the Relational. For the Pre-Structural there is no degree of trust because it always equals 1 since there are not any questions or answers and additionally it is the lowest level and consequently there can be no degree of trust less than 1. Next, having 3 degrees of trust we decide on the final level by taking the average. The average is taken according to the level. Having o 1 for the Pre-Structural, 2 for the Multi-Structural and 3 for the Relational we get:

$$\varepsilon = \frac{1C_1 + 2C_2 + 3C_3}{C_1 + C_2 + C_3},$$

where ε is the final level. ε can be a decimal number. For example if $\varepsilon = 1.5$, then the student is uniformly classified between Single-Structural and Multi-Structural Level.

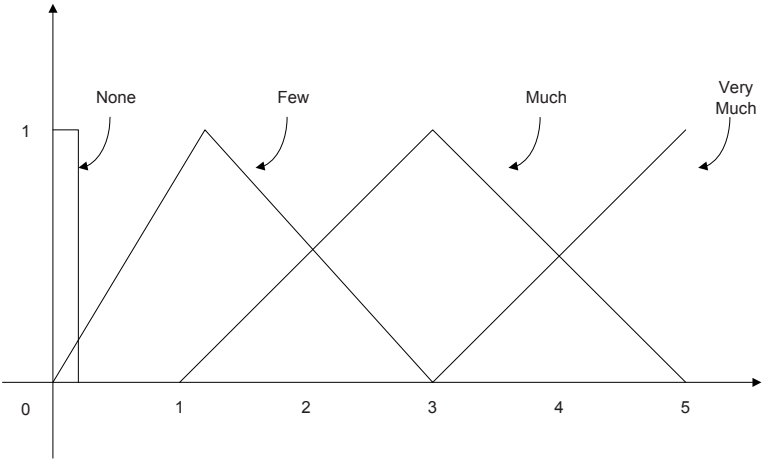


Fig. 4. Number of answers in the specific level

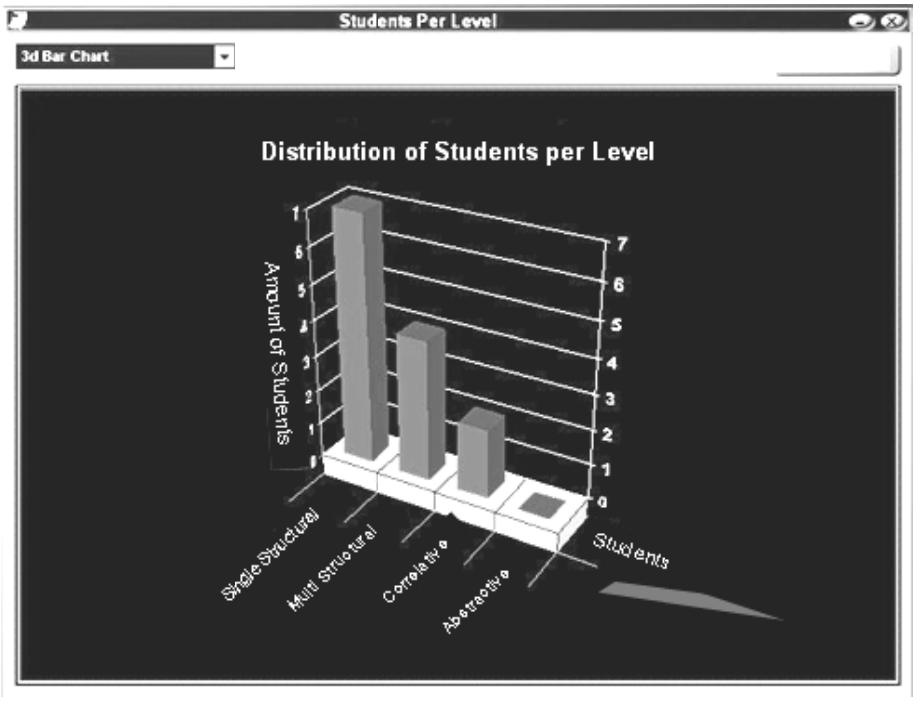


Fig. 5. Students per Level Graph

6 Case Study: The Solo Program

The SOLO program is the interface that contains a powerful intelligent engine that uses an educational diagnostic tool, which basically manages the data of the class and

the students [7]. It is very simple and easy to use, providing help support. Below are stated some selections provided by the interface:

Students per Level: With this selection the user is provided in 2D or 3D graph the distribution on the students depending on the level the students are (Fig. 5).

New Database: This function provides to the user the possibility to create a new database. The window contains combo boxes and textboxes where the user inputs the variables. Using the add button the user inputs a new record to the database. With the delete button the user can delete the present record. By pressing the refresh button the user can refresh the database (for multi-user environment only). With the update button the user can post the database for the changes done, and with the exit button the user closes the window and returns to the main window of the application. By pressing the SaveDB the user saves the database to the hard-disk. (Fig. 6)

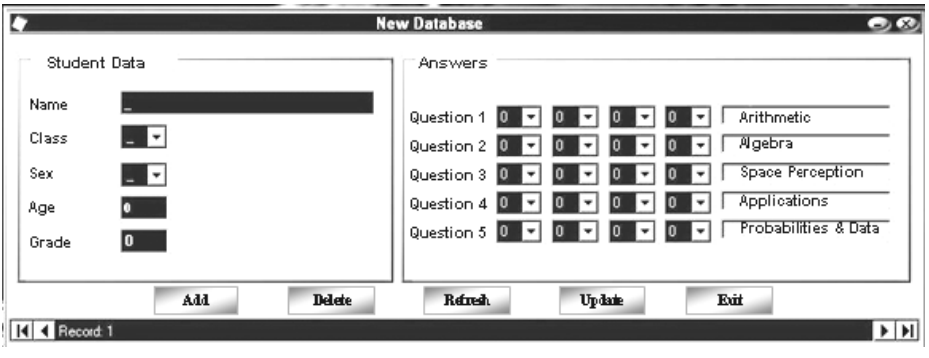


Fig. 6. New Database Window

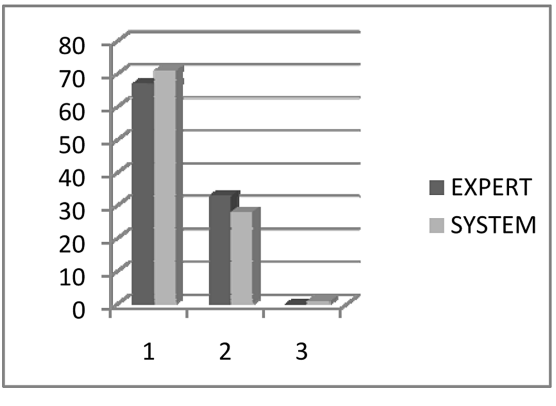


Fig. 7. System -Expert Results Comparison per Level (Dark Grey color corresponds to the expert and Light Gray color corresponds to system)

7 Conclusion

In order to prove the effectiveness of assessment tools, the developed system was applied on 100 high school and senior high school students, and it was tested on mathematics. The correction results obtained by the system were compared to the results obtained by the cognitive science expert. The system's results were found to be very close to the expert results, as it can be seen on the following table (Fig. 7). Concluding, we can say that the assessment tools are trustworthy tools for the educators' cooperation and contribution.

References

1. Biggs, J.B., Collis, K.F.: Evaluating the Quality of Learning: the SOLO Taxonomy. Academic Press, New York (1982)
2. Biggs, J.B., Collis, K.F.: Multimodal learning and the quality of intelligent be- havior. In: Rowe, H.H. (ed.) Intelligence: Reconceptualization and Measurement. Hillsdale, N. J.: Lawrence Erb.freemanm Associates and Hawthorn, Vic.: ACER, pp. 57–76 (1991)
3. Drigas, A., Kouremenos, S., Vrettos, S., Vrettaros, J., Kouremenos, D.: An ex- pert system for job matching of the unemployed. Pergamon-Elsevier Science LTD, Oxford, IDS (2004)
4. Cox, E.: The Fuzzy Systems Handbook: A Practitioner's Guide to Building, Using, & Maintaining Fuzzy Systems (1999)
5. Gogus, O., Boucher, T.O.: A Consistency Test for Rational Weights in Multi- Criterion Decision Analysis with Fuzzy Pairwise Comparisons. Fuzzy Sets and Systems 86, 129–138 (1997)
6. Imrie, B.W.: Assessment for Learning: Quality and Taxonomies. Assessment and Evaluation in Higher Education 20(2), 175–189 (1995)
7. Koleza, E., Barkatsas, A.: The SOLO Taxonomy as a tool for evaluation in Mathe- matics, 2nd Mediterranean Conference on Mathematics Education, Nicosia-Cyprus (2000)
8. Maeda, S., Murakami, S.: The Use of a Fuzzy Decision-Making Method in a Large- Scale Computer System Choice Problem. Fuzzy Sets and Systems 54, 235–249 (1993)
9. Nastoulis, C., Leros, A., Bardis, N.: Banknote.: Recognition Based on Probabilistic Neural Network Models. WSEAS Transactions on Systems 5(10), 2363–2367 (2006)
10. Vrettaros, J., Grigoriadou, M.: Design of hybrid architecture for fuzzy connectionist expert system and its application to approximate student model. In: CATE 96. The first international conference on computers and advanced technologies in education, Cairo (1996)
11. Zadeh, L.F.: Toward a Theory of Fuzzy Information Granulation and its Centrality in Human Reasoning and Fuzzy Logic. Fuzzy Sets and Systems 90, 111–127 (1997)

Author Index