

Petra Perner (Ed.)

LNAI 4597

Advances in Data Mining

Theoretical Aspects and Applications

7th Industrial Conference, ICDM 2007
Leipzig, Germany, July 2007
Proceedings

 Springer

Lecture Notes in Artificial Intelligence 4597

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Petra Perner (Ed.)

Advances in Data Mining

Theoretical Aspects and Applications

7th Industrial Conference, ICDM 2007
Leipzig, Germany, July 14-18, 2007
Proceedings



Springer

Series Editors

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

Volume Editor

Petra Perner

Institute of Computer Vision and Applied Computer Sciences (ibai)
Arno-Nitzsche-Str. 43, 04277 Leipzig, Germany
E-mail: pperner@ibai-institut.de

Library of Congress Control Number: 2007929837

CR Subject Classification (1998): I.2.6, I.2, H.2.8, K.4.4, J.3, I.4, J.6, J.1

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-73434-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-73434-5 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

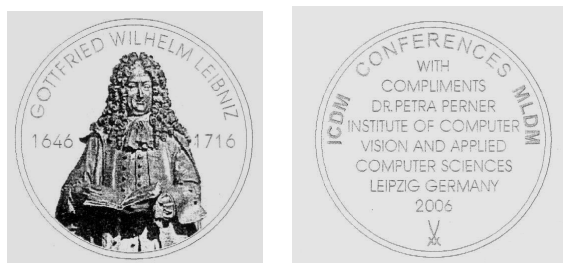
Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2007
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12086436 06/3180 5 4 3 2 1 0

Preface



ICDM / MLDM Medaille (limited edition)
Meissner Porcellan, the "White Gold" of King
August the Strongest of Saxonia

ICDM 2007 was the seventh event in the Industrial Conference on Data Mining series and was held in Leipzig (www.data-mining-forum.de).

For this edition the Program Committee received 96 submissions from 24 countries (see Fig. 1).

After the peer-review process, we accepted 25 high-quality papers for oral presentation that are included in this proceedings book. The topics range from aspects of classification and prediction, clustering, Web mining, data mining in medicine, applications of data mining, time series and frequent pattern mining, and association rule mining.

Germany	9,30%	4,17%	China	9,30%	1,04%	South Korea	6,98%	3,13%
Czech Republic	6,98%	3,13%	USA	6,98%	2,08%	UK	4,65%	2,08%
Portugal	4,65%	2,08%	Iran	4,65%	2,08%	India	4,65%	2,08%
Brazil	4,65%	1,04%	Hungary	4,65%	1,04%	Mexico	4,65%	1,04%
Finland	2,33%	1,04%	Ireland	2,33%	1,04%	Slovenia	2,33%	1,04%
France	2,33%	1,04%	Israel	2,33%	1,04%	Spain	2,33%	1,04%
Greece	2,33%	1,04%	Italy	2,33%	1,04%	Sweden	2,33%	1,04%
Netherlands	2,33%	1,04%	Malaysia	2,33%	1,04%	Turkey	2,33%	1,04%

Fig. 1. Distribution of papers among countries

Twelve papers were selected for poster presentations that are published in the ICDM Poster Proceedings Volume.

In conjunction with ICDM two workshops were run on special hot application-oriented topics in data mining. The workshop Data Mining in Life Science DMLS 2007 was held the second time this year and the workshop Data Mining in Marketing

DMM 2007 was held for first time this year. Right after ICDM, the International Conference on Machine Learning and Data Mining, MLDM 2007, was held in Leipzig (www.mldm.de)

The invited talk was given by Prof. Richter, titled “Case-Based Reasoning and the Search for Knowledge.” The talk illustrated that case-based reasoning on the lower, i.e., more personal levels is quite useful, in particular in comparison with traditional information-retrieval methods.

We saw an increasing number of industrial participants at our conference in the special sessions that covered topics that are important for industry. An invited talk was given by Andrea Ahlemeyer on the topic of “How to Combine Data Mining and Market-Research Technologies?.” A discussion forum that described and discussed the occupational image of the data miner was given by Prof. Gentsch from the Business Intelligence Group Inc. Special talks were given by industry staff in a marketing workshop that described the special problems of different industries.

We are pleased to announce that we gave out the best paper award for ICDM for a second time this year.

We also established an MLDM/ICDM/MDA Conference Summary Volume first the time this year, which summarizes the vision of the three conferences and the paper presentations and also provides a “Who is Who” in machine learning and data mining by giving each author the chance to present himself.

We thank members of the Institute of Applied Computer Sciences, Leipzig, Germany (www.ibai-institut.de) who handled the conference as secretariat. We appreciate the help and understanding of the editorial staff at Springer, and in particular Alfred Hofmann, who supported the publication of these proceedings in the LNAI series.

Last, but not least, we wish to thank all the speakers and participants who contributed to the success of the conference.

July 2007

Petra Pernert

Industrial Conference on Data Mining, ICDM 2007

Chair

Petra Pernert

IBaI Leipzig, Germany

Committee

Klaus-Peter Adlassnig

Medical University of Vienna, Austria

Andrea Ahlemeyer-Stubbe

ECDM, Gengenbach, Germany

Klaus-Dieter Althoff

University of Hildesheim, Germany

Chid Apte

IBM Yorktown Heights, USA

Isabelle Bichindaritz

University of Washington, USA

Leon Bobrowski

Bialystok Technical University, Poland

Marc Boullé

France Télécom, France

Juan M. Corchado

Universidad de Salamanca, Spain

Da Deng

University of Otago, New Zealand

Peter Funk

Mälardalen University, Sweden

Ron Kenett

KPA Ltd., Israel

Eduardo F. Morales

INAOE, Ciencias Computacionales, Mexico

Stefania Montani

Università del Piemonte Orientale, Italy

Eric Pauwels

CWI Utrecht, The Netherlands

Rainer Schmidt

University of Rostock, Germany

Stijn Viaene

KU Leuven, Belgium

Rob A. Vingerhoeds

Ecole Nationale d'Ingénieurs de Tarbes, France

Additional Reviewers

Fabrice Clerot

France Télécom R&D

Francoise Fessant

France Télécom R&D

Carine Hue

France Télécom R&D

Vincent Lemaire

France Télécom R&D

Table of Contents

Invited Talk

Case Based Reasoning and the Search for Knowledge	1
---	---

Aspects of Classification and Prediction

Subsets More Representative Than Random Ones	15
--	----

Concepts for Novelty Detection and Handling Based on a Case-Based Reasoning Process Scheme	21
--	----

An Efficient Algorithm for Instance-Based Learning on Data Streams ...	34
--	----

Softening the Margin in Discrete SVM	49
--	----

Feature Selection Using Ant Colony Optimization (ACO): A New Method and Comparative Study in the Application of Face Recognition System	63
---	----

Outlier Detection with Streaming Dyadic Decomposition	77
---	----

VISRED –Numerical Data Mining with Linear and Nonlinear Techniques	92
--	----

Clustering

Clustering by Random Projections	107
--	-----

Lightweight Clustering Technique for Distributed Data Mining Applications	120
---	-----

Web Mining

Predicting Page Occurrence in a Click-Stream Data: Statistical and Rule-Based Approach	135
Improved IR in Cohesion Model for Link Detection System	148
Improving a State-of-the-Art Named Entity Recognition System Using the World Wide Web	163

Data Mining in Medicine

ISOR-2: A Case-Based Reasoning System to Explain Exceptional Dialysis Patients	173
The Role of Prototypical Cases in Biomedical Case-Based Reasoning ...	184

Applications of Data Mining

A Search Space Reduction Methodology for Large Databases: A Case Study	199
Combining Traditional and Neural-Based Techniques for Ink Feed Control in a Newspaper Printing Press	214
Active Learning Strategies: A Case Study for Detection of Emotions in Speech	228
Neural Business Control System	242
A Framework for Discovering and Analyzing Changing Customer Segments	255
Collaborative Filtering Using Electrical Resistance Network Models	269
Visual Query and Exploration System for Temporal Relational Database	283

Towards an Online Image-Based Tree Taxonomy	296
Distributed Generative Data Mining	307
Time Series and Frequent Pattern Mining	
Privacy-Preserving Discovery of Frequent Patterns in Time Series	318
Efficient Non Linear Time Series Prediction Using Non Linear Signal Analysis and Neural Networks in Chaotic Diode Resonator Circuits	329
Association Mining	
Using Disjunctions in Association Mining	339
Author Index	353

Case Based Reasoning and the Search for Knowledge

Michael M. Richter

Department of Computer Science
University of Calgary, 2500 University Dr.
Calgary, AB, T2N 1N4, Canada
mrichter@cpsc.ucalgary.ca

Abstract. A major goal of this paper is to compare Case Based Reasoning with other methods searching for knowledge. We consider knowledge as a resource that can be traded. It has no value in itself; the value is measured by the usefulness of applying it in some process. Such a process has info-needs that have to be satisfied. The concept to measure this is the economical term utility. In general, utility depends on the user and its context, i.e., it is subjective. Here we introduce levels of context from general to individual. We illustrate that Case Based Reasoning on the lower, i.e., more personal levels CBR is quite useful, in particular in comparison with traditional informational retrieval methods.

Keywords: Case Based Reasoning, Knowledge, Processes, Utility, Context.

1 Introduction

Our starting point is that knowledge and information is some kind of a resource that is used for making processes possible or improving them. Such processes have a certain goal and knowledge is used for achieving it. Therefore knowledge has a certain value; this value is called the utility. In general the utility cannot be defined by looking at the knowledge and the process only. It depends on the specific goal and the person or team performing the process. We refer to it as the context of the user. We distinguish three levels of contexts: A general context that applies to everybody, a group specific context and an individual context. This has an important impact on the problem which knowledge is actually useful.

A major goal of this paper is to compare Case Based Reasoning with other methods to search for knowledge. We show that it is applicable the more specific the context is. For this purpose we discuss the knowledge containers of CBR and pay special attention to similarity, what is the most distinguishing element of CBR. In order to provide relevant knowledge the role of communication between a system and a user is explained and ways towards an optimal dialog is shown. As a final result, we obtain a more sophisticated view under which conditions CBR or related methods can be useful for searching for knowledge.

2 From Knowledge Based Systems to Knowledge Management

Knowledge based systems (also called expert systems) were fully automated systems with a strong logic orientation that could be considered that as ordinary programs, written in a declarative language. One had three more or strictly separated phases: Knowledge and requirement collection, planning and design, and execution. The search for knowledge was done by the system builder and there was no need for the user to do that at application time. Knowledge management was therefore almost not an issue for the user.

Despite many successes of such systems several limitations were known. In particular, the systems were not at all able to handle problems where the strict sequential view had to be given up. These are problem situation that demanded an interleaving of the phases, and one had to start with design and application before the knowledge acquisition was finished. If the system could not solve the problem a human could not help as a “partner”. One of the most important consequences was that human and machine activities should be executed interleaved and concurrently. In engineering disciplines the term “socio-technical processes” was used for such processes, it was later on adapted by computer science. In such processes humans and machines formed a team, they were partners.

The birth of knowledge management was a process, not an event. One observed that they were not simply technical extensions of classical expert systems but demanded the development of systematic investigations. The humans played an essential role in such systems; they had the creativity and the responsibilities. The computer support concentrated (besides the use of pure computations) on providing needed knowledge “at the right time to the right agent in the needed form”.

3 Knowledge, Processes, and Utility

In knowledge based systems one started to regard knowledge as a resource. This was no new insight; it was present long before the electronic age. One could trade, buy, and sell knowledge. Like any other resource, it was needed for some purpose. This purpose was to plan, design, and execute processes. For processes a goal and evaluation methods need to be assigned. On this basis it is possible to measure the success and the improvement of the process performance when using some resource.

3.1 Processes and Utilities

Processes have goal, i.e., a utility in economical terms. Utility has a relational and a functional form. The relational form is a preference relation and the functional form a real valued function. The preference relation “b is preferred over a” is denoted by $a < b$; $a \leq b$ says that a is not preferred over b and $a \sim b$ denotes indifference.

Utility functions u operate on actions, whole processes or decisions. They assign real numbers as values to the elements of their domain A :

$$u: A \rightarrow \mathfrak{R}.$$

A utility function always induces a preference relation by:

$$b \text{ is preferred over } c \Leftrightarrow u(b) > u(c).$$

Both, utility functions and preference relations are usually complex. User preferences are easier to acquire than utility functions. In Mathematics, utility theory is an established discipline. Among the mathematical approaches the most prominent one is the von Neumann-Morgenstern theory [12]. However, it is an old insight that often utilities are not formulated in mathematical terms. Utility is rather subjective because it depends on the special situation of the person or the company. The term “subjective” is not used here in the sense of psychology; it just means that there is no model of the utility visible to the outside. A detailed discussion of subjective expected utility has been given in [19], which was a highly influential book. The rational behaviour of humans is expressed in the equation

$$\text{subjective value} = \text{subjective probability} \times \text{subjective utility.}$$

The subjective value has to be maximized. The subjective view becomes in particular important if knowledge is needed. The utility of knowledge is strongly influenced by the goal and by the knowledge and level of understanding the user has.

3.2 Processes and Knowledge

The knowledge and information units needed to perform a process are called the *info-needs* [9]. They are stored in *info-sources* as shown in Figure 1.

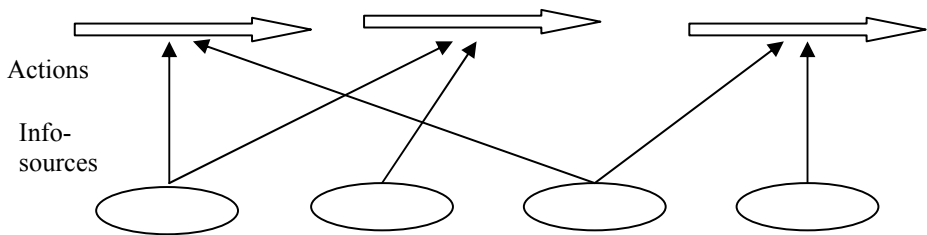


Fig. 1. Processes, info-needs, info-sources

The agent to take care of this is usually called the knowledge manager who has two tasks:

- 1) To define the info-sources, to structure them, to fill them, and to apply maintenance.
- 2) To take care that the agent performing the process obtains the info-needs in the way needed, i.e., in the right form and at the right time.

Structuring and searching for knowledge is not for free. It has costs that can be measured in terms of money, time, inconvenience, and other units. Some financial figures are given in [10].

However, there is not always such a manager and often this agent has limited abilities. Hence, the acting agent has to search on its own for the needed info-sources.

There are basically two ways to provide knowledge. The first one is on demand. That means, a question is presented and an answer is returned. Here the answering

agent knows that some knowledge is missing. Nevertheless, often the answer is not very satisfactory. This is mainly due to the fact that the query formulation is incomplete, misleading, or not understandable. The second way is pro-active. Here the knowledge manager has to act on her own; the addressee may not even know that certain knowledge is needed. In many situations, both ways are combined. For instance, a user formulates a query where not only some answer is given but additional information is provided that is useful for the user.

What has to be done is to bridge the gap between the info-needs and the knowledge delivered. As we will see, the width of that gap can be measured by a similarity measure; this is discussed in the section 5.

What has to be observed is that often knowledge search takes place when no knowledge manager is present. For instance, if I am in the process of downloading my program, who is telling me pro-actively that there is a program from another company that is a serious alternative? The best that can happen is that my company has an internal management that supports me. In addition, often it is not clear who searches for the knowledge: The one who needs it or the one who has it? This will be discussed in section 5.3 on communication.

3.3 Specifications and Knowledge Search

Knowledge search is a process that can be performed interactively or fully automatic. In order to measure the success a specification has to be given. This specification is the intended utility. As discussed above, the utility can be defined in a formal, mathematical way, or informally. The formal specification allows in principle a verification as in programming languages. Mostly, however, utilities will be presented informally; in such cases formal proofs have to be replaced by informal arguments. In [16] we have described this approach for similarity based search in more detail.

3.4 Contexts and Their Levels

In principle, there is usually an infinite amount of knowledge that has a relation to the process of interest. What is actually needed depends on the context in which the process takes place. There are various ways a goal can be missed, for instance if the knowledge is incomplete, too general, confusing, or not understandable. We distinguish external and internal contexts. The external context considers what happens around the performing agent (the specific task, the general circumstances etc.) The internal context is concerned with the knowledge and experience of the agent, its preferences etc. The context can be more or less general.

We define three *context levels* as shown in Figure 2.

- 1) The general level: Everybody has the same context, for instance, when one searches for the Lufthansa schedule.
- 2) The group level: There are groups of users and each group has a different context; for instance, different social groups look for different entertainments.
- 3) The individual level: The context depends on the user, for instance, when one searches for an employee with specific abilities.

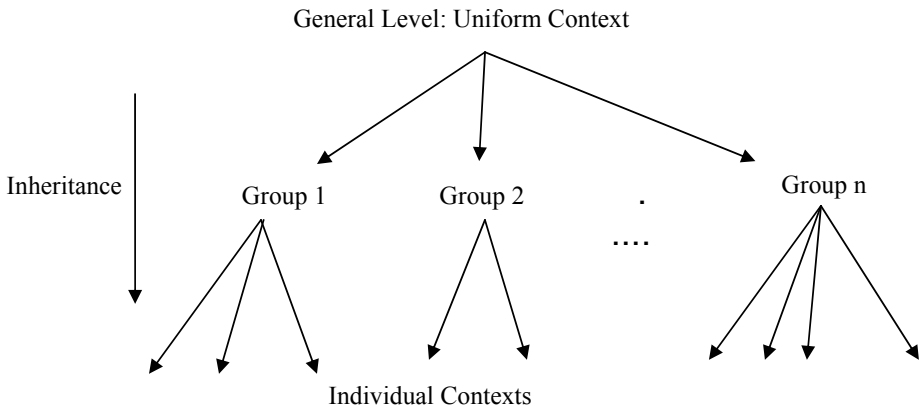


Fig. 2. Context Levels

Ordinarily, everybody has utility aspects from all three levels; one is an individual, belongs to one or more groups and shares also some general views. On the general level often one finds mathematical utility function; the more one goes down the more subjective the utilities play a role and the more relevant the internal context will be. There are two major problems associated with contexts:

- a) These contexts are not static, they rather change over time. The speed of change increases the more one comes down to the lower levels; on the general level the change proceeds very slowly. Each change has to be reflected in a system that provides knowledge; this is known as the maintenance task.
- b) The contexts are partially unknown. This is a little problem only on the general level. On the group level it requires, depending on the group, sometimes much effort to acquire it. On the individual level there are mainly two possibilities. The first one is to learn preferences from user's histories; what necessitates that these are recorded. The second possibility is a direct communication with the user,

At the general level general search machines and retrieval techniques are located, like those that search in the web. The more one goes down the levels the more difficulties one has with general machines. The recent activities on level three run under the name *personalization and context awareness*. This means, one is not only task-centric but at least as much user-centric.

4 Case Based Reasoning

Case Based Reasoning (CBR) is now an established technology. We start with a short introduction into the basic concepts. They are all concentrated around search for knowledge and it is justified to call CBR a knowledge search technology. An overview over CBR and knowledge management is given in [3].

4.1 Experiences

The original and basic intention of CBR was to use previous experiences for solving actual problems. Experiences, if stored, are an important part of knowledge. The idea of *case* is a recorded episode, where a problem or problem situation was totally or partially solved. In its simplest form a case is represented as an ordered pair (problem, solution). In order to establish such a case it suffices that the corresponding episode happened in the past. A case base or an experience base is a set of recorded experiences [2].

In order to use such a case base there is a simple and convincing principle based on what one calls analogy: If there is a new problem that is closely related or *similar* to the problem description of a recorded case, then make use of the latter.

The basic scenario for CBR looks from a naive point of view as follows: In order to find the solution to an actual problem one looks for a similar problem in an experience base, takes the solution from the past and uses it for finding a solution to the actual problem. This is shown in the Aamodt-Plaza-Cycle [1] in Figure 3. The cycle describes the CBR activities only superficially and in the sequel we will discuss several aspects in more detail.

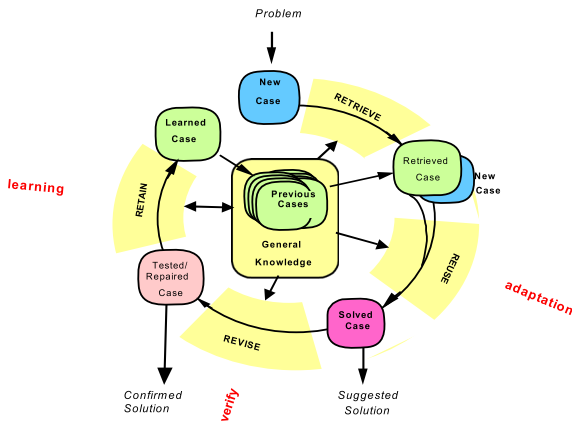


Fig. 3. Aamodt-Plaza Cycle

When the problem is presented to the system a search in the case base takes place. This search is more involved than a data base search because one is not looking for a specific object but rather for a “relatively useful” one. The formal concept to do this is the similarity measure that selects the most similar problem (the nearest neighbour); this is discussed in section 5.

The retrieved case is then reused. This does not mean that it is used directly as retrieved because the old solution may not quite do the job because of the difference of the old and the new problem. To take care of this, some adaptation takes place.

The solution obtained in this way is tested and verified. If the solution passes the tests, then one has a new experience that can be stored in the case base. This means, a learning step took place.

CBR as introduced so far is some kind of *experience mining*. Many successful applications of CBR have been done in this area. The approach has been extended to more general situations where experiences are recorded. Major examples are the experience factory [2] and the technique *lessons learned* [23]. The approaches differ technically but aim at the same goal.

4.2 Question Answering and General Search for Knowledge

The applications of CBR have been generalized extensively. For this, it was necessary to extend the notion of similarity. It had not only to compare old and new problems, but much more general objects. Therefore similarity measures played the role of “*partnership measures*“. The intention of partnership is that both objects cooperate more or less well as partners. Because the possible partners may come from different sets the similarity measure has to compare quite different objects. In the extended view, the measure compares objects like the ones seen in Table 1. This view is presently dominating and the term “case base” from CBR is often replaced by expressions like product base, document base, etc.

Table 1. General partners

Info-needs	Info-Sources
Knowledge needed	Documents
Questions	Answers
Functionalities	Machines
Desired products	Available products

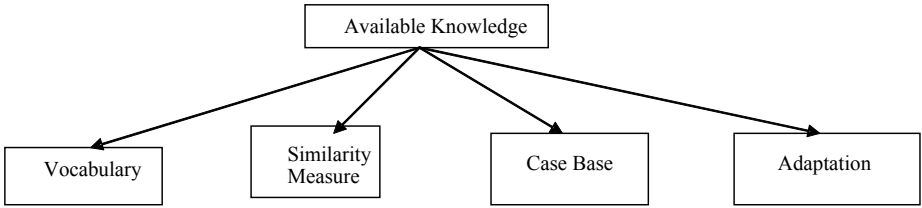
The similarity measure is a way to measure the distance between the objects of interest numerically. The nearest neighbour search then bridges the gap between these objects, for instance, between info-needs and info-sources. The main difficulty is to perform the search context oriented.

A crucial point is also, that, despite the widening of the applications, the essentials of the CBR technology could still be used. Moreover, even the same tools could be applied. An example is CBR-Works [8] from empolis and its extension orange [13].

An example on the group level context is the Simatic *Knowledge Manager*, a comprehensive industrial application developed by empolis. It provides online support and self-service for the group of customers and technicians of Siemens Company. The system is integrated in the call center of the company. The queries can be formulated in natural language

4.3 Case Based Reasoning and Its Knowledge Containers

A convenient way to describe Case-Based Reasoning is to introduce the concept of *knowledge containers* [15, 17]. Knowledge containers are not sub modules of a system, because they do not solve any sub problem. They rather are description elements that can be filled with knowledge units. In CBR we identify four major knowledge containers as shown in Figure 4.



There is an interaction between the containers:

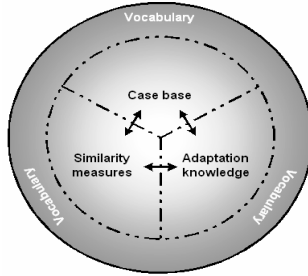


Fig. 4. Knowledge Containers

The containers are not simply empty barrels to be filled. The opposite is the case; they are equipped with a partially complex structure.

The importance of the vocabulary container is the same as in any knowledge based system: What has no name cannot be discussed. In order to be convenient for the user each term can be enriched by an explanatory record, the term record (see [14]). The term record can contain synonyms, quasi-synonyms, advises when and how to use, sources etc.

The case base is, mainly for retrieval purposes, usually equipped with additional structures. Structured case bases run also under the name *case memory*.

The similarity container and the adaptation container are discussed below in section 5.

All the containers are strongly related to each other and the knowledge can be shifted between the containers. This gives CBR systems a large degree of flexibility. It allows to employ learning methods and to keep up with context changes.

The containers play a role when a CBR system is maintained or constructed; then the containers have to be filled. We distinguish two phases:

- a) The planning (or compilation) phase: That contains everything that happens before an actual problem is presented;
- b) The run time phase.

For ordinary programs and knowledge based systems, one has to understand all knowledge that enters the system. In a CBR system, the cases need not to be understood at compile time, they are just filled into the case base container. Understanding the cases is only necessary at run time when they are actually used. This has the advantage that one can start with a system that works somehow, but not necessarily very well. At a later time the system can be improved.

The construction of the measure is called similarity assessment; the measure has to be understood at compile time. This is obvious because it has to reflect the goals and the context. But here also we make use of the flexibility: One can start with a very general context and later on make it more specific.

When going down the hierarchy of context levels all knowledge containers have to be adapted. For instance, individuals have a special vocabulary; terms not of interest should be removed. Adaptation needs are personal and rules should be corrected.

5 Similarity Measures and Adaptation Rules

Both, adaptation rules and similarity measures are of particular interest. Similarity is by far the most important concept in CBR; adaptation rules provide a number of difficulties.

5.1 Similarity Measures

We introduce some notation for standard concepts. Similarity measures sim are defined on ordered pairs:

$$\text{sim}: U \times U \rightarrow [0, 1] \text{ or } \text{sim}: U \times V \rightarrow [0, 1]$$

for sets U and V . For simplicity we assume that all objects from U and V are represented as attribute-value vectors.

The role of a and b in $\text{sim}(a, b)$ is not symmetric; usually a is a given object (called the *query object*) and b is a possible *answer object*.

In a sales situation the customer sends a demand to the shop and the query objects are the customer demanded products and the answer objects are the available products. Hence we have $V \subseteq U$. In information retrieval U and V are different. In more general situations like document search U and V may be disjoint.

We denote the nearest *neighbour relation* by $\text{NN}(a, b)$, stating that b is a nearest neighbour of a . A CBR system performs two types of computations:

- Computing the similarity value between objects a and b .
- Searching for the nearest neighbour(s).

During the search the nearest neighbour is approximated and in each search step a similarity computation is done.

5.2 Adaptation Rules

The rules in the adaptation container take care of changing the retrieved objects for proper use. The situation is that despite the similarity of the answer and the query object, both are different. A serious problem arises if there are very many rules that can be applied. This occurs, for instance, when one searches in catalogues. Often only less than one percent of the available products are listed explicitly; all others are obtained from them by modifications or additions. This excludes in the first place a simple nearest neighbor search because the measure is not informed about the adaptation rules. On the other hand, the adaptation rules do not know anything about

the intended utility and similarity. What one would like to have is a measure that takes the utility of a product after adaptation into account.

An approach to obtain such a measure is described in [22]. It uses genetic algorithms for learning the measure from user feedback. It extends the technique in [22] to learn local similarity measures. This is a non-trivial example of shifting knowledge from the adaptation container to the similarity container.

5.3 Retrieval and Communication

Retrieval is in the first place just search. This view is in so far insufficient as the query may initially not be formulated completely and precisely. Both, the user and the knowledge provider are cooperating in this respect. They are both trying to dig out the aspects relevant for the user and the context; this requires a communication. Such *personalization* efforts play presently a big role in recommender systems. They are, among others, intended to present some choice of a set of alternatives of decisions to the user.

As an example for communication problems in the form of a dialog we consider a sales process. In electronic commerce the seller is presented as a web site and this seller gets an initial query from the buyer that needs to be completed. There are a number of conditions for a successful sales talk. The main ones are that it is understandable to the user and it is short. The danger for the web site is that the buyer may become annoyed and will quite before a sale took place. Hence, the dialog has to focus on two aspects simultaneously: To obtain more insight into the context of the user and to come closer to the optimal product to sell.

Several CBR approaches to automated sales dialogs have been suggested so far [20]. The approaches have in common that there is a list of possible questions to ask the user that are presented in the form of attributes where the user has to fill in some value. One goal is to ask a minimal number of questions, i.e., to reduce the dialog length. For this purpose, questions will be selected according to their relevance for the customer's utility function. Initially, this is only partially known and has to be completed during the dialog. It is guided by some selection mechanism that selects the next question. This choice is dominated by getting the most information for the intended product. One way is, to make use of ordinary entropy. This, however, neglects the fact that some aspects of the utility of the user are already known. A refined view says that a question leads to a new dialog situation s in which all products that do not exactly match the current attribute value are excluded. This view leads to the information content shown in (1), where A is the selected attribute.

$$extgain(s, A) := - \sum_{v \in \{v_1, \dots, v_m\}} p_v \cdot \log_2 \left(\frac{\sum_{c \in C} sim(A(c), v)}{|C|} \right) \quad (1)$$

In orange 3.1, a new attribute selection method was introduced [13]; it is essentially a slight extension to *extgain*. A further refinement is using the variance of similarities. Higher similarity variance in the set of candidate cases implies better discrimination between possible target cases (products) and unsatisfactory. The *simVar* measure is calculated as shown by (2), where v is a value of attribute A .

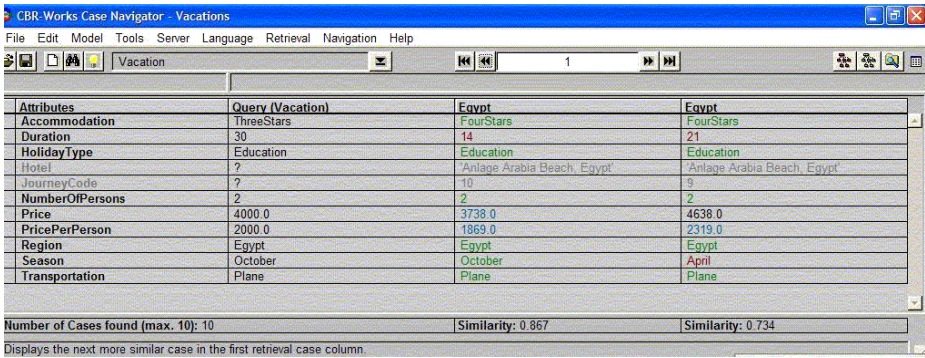
$$\text{simVar}(q, A, v, C) := \frac{1}{|C|} \cdot \sum_{c \in C} (\text{sim}(q_{A \leftarrow v}, c) - \mu)^2 \quad (2)$$

Here we consider the transition of a dialog situation s to the new situation s' by assigning the value v to the query attribute A of query q (denoted $q_{A \leftarrow v}$); μ is the average value of all similarities, and C again the case base.

Statistical evaluations suggest that among these (and some other) methods there is no single best method for all scenarios. In most experiments *simVar* has outperformed *extgain*, although there is yet no theoretical foundation for that.

5.4 Example

CBR-Works provides a user friendly dialog box as shown in Figure 5. The user gets a question (in natural language for providing the wanted value of some attribute that is chosen as described in 5.3. For this, the user is offered the possible alternatives, for instance, by providing a list or a numerical interval. The attributes are weighted and the weights can be changed by the user. If needed, adaptation can take place. The adaptation rules are invisible by the user. The screen shot shows the two nearest neighbors but more alternatives can be offered on demand. There is also the possibility to show details if wanted so.



Attributes	Query (Vacation)	Egypt	Egypt
Accommodation	ThreeStars	FourStars	FourStars
Duration	30	14	21
HolidayType	Education	Education	Education
Hotel	?	'Anlage Arabia Beach, Egypt'	'Anlage Arabia Beach, Egypt'
JourneyCode	?	10	9
NumberOfPersons	2	2	2
Price	4000.0	3738.0	4638.0
PricePerPerson	2000.0	1869.0	2319.0
Region	Egypt	Egypt	Egypt
Season	October	October	April
Transportation	Plane	Plane	Plane

Number of Cases found (max. 10): 10 Similarity: 0.867 Similarity: 0.734

Displays the next more similar case in the first retrieval case column.

Fig. 5. CBR-Works User Interface

The dialog box deals with selling vacations. The user can see where the demands are matched exactly and where not. On the bottom one can see the computed similarities between the query and the offer.

6 CBR, Information Retrieval, and Knowledge Extraction

Both, information retrieval and knowledge extraction from texts are concerned with documents. Information retrieval is mostly searching for documents as a whole, while feature extraction searches in one or several documents for useful information.

6.1 Information Retrieval and Case Based Reasoning

In this section we compare the traditional vector space model of Information Retrieval (IR) with similarity retrieval in the sense of CBR. Both approaches use similarity measures for describing the usefulness of a document.

In Information Retrieval, the access to a document is done by looking at terms that have been already extracted from the document. The extracted terms form the bridge between the query and the document. Both use the same vocabulary, namely some terms of interest. Which terms are chosen can be based on some knowledge about the domain and the intended user. These terms are compared with the terms in the query.

Each selected term is associated with a term weight that is supposed to reflect the importance of the term. This gives rise to an m -dimensional weight vector that is responsible for the document retrieval. The advantage is that this does not require any additional knowledge. On the other hand, group specific knowledge cannot be entered. A standard approach uses weights on the basis of the term frequency, f_{ij} (the number of occurrence of term y_j in document x_i ; and the inverse document frequency, $g_j = \log(N/d_j)$, where N is the total number of documents in the collection and d_j is the number of documents containing term y_j .

The documents weights w_{ij} and the query weights v_j are

$$w_{kj} = f_{kj} \cdot \log(N/d_j)$$

and

$$v_{ki} = \log(N/d_j) \text{ if } y_j \text{ is a term in } q \text{ and } 0 \text{ otherwise.}$$

This gives rise to a document vector d and a query vector v that have to be compared. The involved similarity measures for the comparison are not very sophisticated; usually the *cosine* of a query q and a document d is taken. There are two main criteria of success: Recall and precision, measuring the ratio of the number of relevant records retrieved to the total number of relevant records in the database. These criteria are natural on the general level and not much more can be done there. On the group level they can, however, be criticized because they do not take into account that the recall and precision may be restricted to the documents and queries of the special context and goal of the user.

There are some differences between the vector space approach and the attribute-value representation in CBR. In CBR, the attributes have a domain from which the values are taken. In IR, the vector coordinates are simply numbers; there are no variables that could be instantiated and each coordinate is labelled by a fixed term. In CBR, the weights are parts of the measure; in IR, they are parts of the object representation. It is, however, not difficult to unify the approaches. For this purpose, we introduce for each term an attribute $\text{Frequency}_t(\cdot)$ and take the weights as in a weighted Euclidean measure. This measure is equivalent to the measure in the vector space model. Although the principle equivalence, IR is more suitable on the general level while CBR aims at more specific levels.

If the retrieval process is not convincing then both, CBR and IR apply adaptation. The difference is that in CBR the solution is adapted while in IR rather the query is rewritten (because one cannot rewrite a document). A widely used method is query expansion. Here the original query is supplemented with additional terms. The idea is

to add such terms to the query that are similar the given ones. This means that the newly introduced terms are close to the old ones with respect to some similarity measure. Automated query expansion uses a frequency measure on pairs of words; i.e., counting how often the words occur together in texts.

This can be based on user feedback, inspection of past optimal queries, or others. In addition, the term weights can be improved; for example; by using statistical information. To some extent query expansion takes care of the user's needs, but only in a very limited way. More refined ways like employing a thesaurus is rarely used.

6.2 Information Extraction from Text

Often texts contain useful but hidden information. This happens, for instance, in domains like law, science, medicine etc. For retrieval, IR methods are helpful but rather restricted, like extracting terms. On the other hands, CBR methods are also not directly applicable because the cases are recorded as text. To use them directly, considerable case engineering effort is needed. To overcome this difficulty, natural language processing methods (NLP) seem to be necessary.

An example where this was carried out is the system SMILE, see [7]. This system uses the standard NLP based information extraction system Autoslog [18]. It extracts various kinds of information from the text like certain facts and syntax related information for formulating queries. The cases were legal cases.

A related system on the basis of CBR is jCOLIBRI, [5,11], where a generalizations and learning was applied.

7 Conclusion

We described Case Based Reasoning as a knowledge search technology. From this point of view CBR is competing with many other such technologies and the question arises, which one to choose in which situation. There is no general solution to the problem finding an optimal method. One can formulate, however, some directions. For this purpose we introduced levels for contexts. On the general level, existing search machines are superior. The group level occurs often in companies, and there much more expert knowledge is needed. Typical examples of the individual level are provided by e-commerce. Here we can observe a mixture of general search machines and personalization.

References

1. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications* 7, 39–59 (1994)
2. Althoff, K.-D., Althoff, B., Althoff, B.A., von Wangenheim, C. G., Tautz, C.: CBR for Experimental Software Engineering. *Case-Based Reasoning Technology*, 235–254 (1998)
3. Althoff, K.-D., Weber, R.O.: Knowledge Management in Case-Based Reasoning. *Knowledge Engineering Review* 20(3), 305–310 (2005)
4. Ardito, R., Bara, B.G., Blanzieri, E.: A cognitive Account of Situated Communication *COGSCI 2002* (2002)

5. Bello-Tomás, J.J., González-Calero, P.A., Díaz-Agudo, B.: JColibri: an Object-Oriented Framework for Building CBR Systems. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, Springer, Heidelberg (2004)
6. Bergmann, R., Richter, M.M., Schmitt, S., Stahl, A., Vollrath, I.: Utility-Oriented Matching: A New Research Direction for Case-Based Reasoning. In: Vollrath, I., Schmitt, S., Reimer, U. (eds.) Proc. of the 9th German Workshop on Case-Based Reasoning, GWCBR'01, Baden-Baden, Germany, Baden-Baden, Germany. In: Schnurr, H.-P., Staab, S., Studer, R., Stumme, G., Sure, Y. (Hrsg.): Professionelles Wissensmanagement. Shaker Verlag (2001)
7. Brüninghaus, S., Ashley, K.: The Role of Information Extraction in Textual CBR. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI) (SNLAI), vol. 2080, pp. 74–80. Springer, Heidelberg (2001)
8. CBR-Works (2003), www.ualberta.ca/courses/SENG/609.13/W2004/06.%20CBR-Works.pdf
9. Holz, H.: Process-Based Knowledge Management Support for Software Engineering, Doctoral Dissertation University of Kaiserslautern, dissertations.de Online-Press (2002)
10. Jacobson, A., Prusak, L.: The Cost of Knowledge. Harvard Business Review (2007)
11. jcolibri (2002) <http://gaia.fdi.ucm.es/projects/jcolibri>
12. von Neumann, J., Morgenstern, O.: Theory of Games and Behavior, 1953th edn. Princeton University Press, Princeton, NJ (1944)
13. orange:dialog. In: orange: Open Retrieval Engine 3.2 Manual. empolis – knowledge management, <http://www.km.empolis.com/>
14. Richter, M.M.: Terminology in Complex Domains. In: Bock, H., Polasek, W. (eds.) Proc. Of the 19th Annual Conference of the Gesellschaft für Klassifikation. Studies in Classification, Data Analysis and Knowledge Organization, pp. 416–426. Springer, Heidelberg (1995)
15. Richter, M.M.: Introduction. In: Lenz, M., Bartsch-Spörl, B., Burkhard, H.-D., Wess, S. (eds.) Case-Based Reasoning Technology. LNCS (LNAI) (SNLAI), vol. 1400, Springer, Heidelberg (1998)
16. Richter, M.M.: Foundations of Similarity and Utility. In: Proc. FLAIRS07, AAAI Press, Stanford, California (2007)
17. Richter, M.M.: Similarity. In: Perner, P. (ed.) Case-Based Reasoning on Signals and Images, Springer, Heidelberg
18. Riloff, E.: Automatically Extraction Information Patterns from Untagged Text. In: Proc. of the 13th National Conference on Artificial Intelligence, AAAI Press, Stanford, California (1996)
19. Savage, J.L.: 1954 Foundations of Statistics, 2nd Rev. edn. Dover Publications, Mineola (Reprint) (1972)
20. Schmitt, S., Bergmann, F.R.: A formal approach to dialogs with online customers. In: 14th Bled Electronic Commerce Conference (2001)
21. Schmitt, S., Dopichaj, P., Domínguez-Marín, P.: Entropy-based vs. Similarity-influenced: Attribute Selection Methods for Dialogs Tested on Different Electronic Commerce Domains. In: Craw, S., Preece, A.D. (eds.) ECCBR 2002. LNCS (LNAI), vol. 2416, Springer, Heidelberg (2002)
22. Stahl, A.: Learning of Knowledge-Intensive Similarity Measures in Case-Based Reasoning. Kaiserslautern (2003)
23. Weber, R., Aha, D.W., Becerra-Fernandez, I.: Intelligent Lessons Learned Systems. Expert Systems with Applications 20(1), 17–34 (2001)

Subsets More Representative Than Random Ones

Ilia Nouretdinov

Department of Computer Science
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England
`ilia@cs.rhul.ac.uk`

Abstract. Suppose we have a database that describes a set of objects, and our aim is to find its representative subset of a smaller size. Representativeness here means the measure of quality of prediction when the subset is used instead of the whole set in a typical machine learning procedure. We research how to find a subset that is more representative than a random selection of the same size.

1 Introduction

Let us have a training set $Z = \{z_1, \dots, z_n\}$ and a testing set $Z_T = \{z_{n+1}, \dots, z_{n+k}\}$, $z_i = (x_i, y_i) \in X \times Y$ where $X = \mathbb{R}^m$ and $Y = \mathbb{R}$.

Suppose $n' < n$ and our aim is to choose a subset $Z' \subset Z$ consisting of n' elements. We call this operation *subset selection*. We like the most essential information about data Z to be saved in Z' .

Suppose we are studying a data set, which is too large to process it directly. To make some conclusion about it, one may take random selection. Such subset is expected to have similar distribution and properties. We can ask the following question. Such selection is preferred to be random, because non-random selection is usually less representative. But a random selection has only random level of representativeness, not high one! If there are non-random selections, which are less representative than random, then we can expect some other non-random selections to be more representative than random.

So, theoretical point is looking for subsets which are more representative than random. The practical motivation is potential time economy in the empirical comparison procedure.

This work is based mainly on experimental results, but there is some theoretical intuition behind them. To exclude a small amount of typical elements leads to smaller loss of information than to exclude a small amount of marginal elements; but this is not as clear for a larger exclusion.

The idea of our approach is following. We start with a large family of tests for randomness. Other applications of such test for machine learning problems are discussed in [1-4].

Based on this, so-called total randomness deficiency is defined as a function of a set and its element, which is a measure of how this element is informative in comparison to others.

First, we define a method of compression a set of size q into its subset of size $(q - \lfloor \log(q) \rfloor)$, by excluding less informative elements. Next, we compress a set of size q^2 into its subset of size q , using $q \rightarrow (q - \lfloor \log(q) \rfloor)$ compression for different subsets, and some voting procedure.

Experimental check was done on the Boston Housing Dataset. It shows that a selection of subset done by our method performs better than 95% random selections of same size. Additionally, it is almost as good as the whole dataset.

Practically, this method could be useful for the economy of time for comparison different machine learning methods on a training set before applying them to a test set.

In the experimental model of this work, comparison of different machine learning methods is similar to the well-known problem of feature selection (or dimension decrease). We use a naive method of such selection itself, but a specific step is added to this. The samples can be understood (by duality) as features of features, so feature selection can be naturally preceded by feature of feature selection, that is equivalent to our compression of a set into its representative subset.

To sum, this work consists of two parts. First, we describe our method of set compression. It is mainly based on the notion of test for randomness.

The second part is an experimental check. We formulate a practical criterion of representativeness, and then ensure on a real database that Z' fits this criterion better than a random selection of the same size.

2 Randomness Theory Background

2.1 Tests for Randomness

Definition 1. Let U be a finite set, $A \subset U$, $z \in A$, $I \in \{1, \dots, N\}$

$$f : (I, A, z) \rightarrow [0, 1]$$

$$\frac{|\{z \in A : f(I, A, z) < \gamma\}|}{|A|} < \gamma$$

$$I, A \quad 0 < \gamma < 1$$

Definition 2. Let f be a function defined on (I, A, z) for $z \in A$.

$$\mu_f(z|A) = \sum_I (-\log f(I, A, z)).$$

2.2 Example: Test Based on a Method of Prediction

In the papers [2,3] the concept of test for randomness is used as a way of transformation of a method of bare prediction into a method of confident multi-prediction via a test for randomness based on a bare prediction method.

For the current work, we do need all the information for the approaches described in these papers, except the correspondence between methods of prediction and tests for randomness.

So, we need to construct a test for randomness based on a method of bare prediction. Details may be different according to a specific method, but in any case we need to define a measure of strangeness that is a measure of disagreement between a set and its element. For Nearest Neighbour method, disagreement between z_i and $\{z_1, \dots, z_n\}$ is

$$\alpha_i = \frac{\min_{j:j \neq i, y_j = y_i} d(x_i, x_j)}{\min_{j:j \neq i, y_j \neq y_i} d(x_i, x_j)} \quad (1)$$

where d is a metric on X .

Let us have a family of different distances d^1, \dots, d^N . For nearest method, the test for randomness is

$$f(I, A, z_i) = \frac{|\{j \mid \alpha_j^I \geq \alpha_i^I\}|}{|A|}. \quad (2)$$

where

$$\alpha_i = |y_i - (y_j \mid j \neq i, \text{dist}^I(x_i, x_j) \rightarrow \min)| \quad (3)$$

For a method another than Nearest Neighbours, the same idea may work: α_i should be larger if the sample $z_i = (x_i, y_i)$ somehow contradicts the method applied to the whole set $\{z_1, \dots, z_n\}$, e.g. leave-one-out prediction for it is another than its label.

It can be checked (see e.g. [2]) that this function satisfies the definition of a family of tests for randomness.

3 Compression

We base on two ideas: (1) there is a natural correspondence between methods of predictions and tests for randomness; (2) if an example is untypical according to most tests then it is more informative for comparison of corresponding methods of prediction.

3.1 Basic Operation

Recall the formula of total randomness deficiency:

$$\mu_f(z|A) = \sum_I (-\log f(I, A, z)). \quad (4)$$

The family f is variable. We will detail it when describing our experiments.

Now we define the operation $F_r : A \rightarrow B$ deleting r the most typical elements $z \in A$ for which $\mu_f(z | A)$ is minimal. It is desirable the quantity of such deleted elements be small, otherwise it can affect the distribution too much. This is why we use $r = \lceil \log |A| \rceil$ and do not use basic operation F_r in original form to compress large data sets into little ones.

3.2 Next Step of Compression

Let q be a prime number and $|A| = q^2$. Our aim is compression $|A|$ into the set of size q .

Fix a numeration of its elements with two indices:

$$A = \{a_{i,j} \mid i, j \in GF_q\} \quad (5)$$

where GF_q is a finite field of the size q .

Choose $q^2 + q$ q -element subsets of A as follows:

$$A_u = \{a_{i,j} \mid i = u\} \quad (6)$$

$$A_{u,v} = \{a_{i,j} \mid j = u + iv\} \quad (7)$$

for $u \in GF_q, v \in GF_q$. Such sets hold the following simple properties:

- each $a \in A$ belongs to exactly $q + 1$ different $A_{u,v}$;
- each pair $\{a, b\} \subseteq A$ is a subset of exactly one $A_{u,v}$.

These properties allow running of a voting procedure. Recall that each $F_r(A_{u,v})$ is a subset of $A_{u,v}$ of the size $q - \lceil \log q \rceil$. For each element $z \in A$ let $w(z)$ be the number of different pairs (u, v) such that $z \in F_r(A_{u,v})$. Let $B = F(A)$ consist of q elements with largest $w(z)$. This is called $F : A \rightarrow B$ operation with $|A| = q^2$ and $|B| = q$.

4 Experimental Check

4.1 Data Set

Experiments are done on the Boston Housing Dataset. It contains $n = 401$ training and $k = 105$ test examples, $m = 13$ scalar attributes, and the price (scalar) as a label.

After a linear transformation (extraction of mean value and division by mean square deviation), each attribute has mean value 0 and deviation 1 on the training set. Linear transformation with same coefficients is also applied to the testing set.

4.2 Family of Tests

Let

$$f(I, A, z_i) = \frac{|\{j \mid \alpha_j^I \geq \alpha_i^I\}|}{|A|}. \quad (8)$$

with alpha function:

$$\alpha_i = |y_i - (y_j \mid j \neq i, \text{dist}^I(x_i, x_j) \rightarrow \min)|. \quad (9)$$

It can be checked that this function satisfies the definition of family of tests for randomness.

We suppose $I = 1, \dots, N = 2^m - 1$ is a numeration of all subsets of the set of 13 attributes and dist^I is the Euclidean distance calculated when only to the attributes from I -th set are used (feature selection problem). The total randomness deficiency is

$$\mu_f(z|A) = \sum_I \log f(I, A, x). \quad (10)$$

4.3 Step of Compression

As mentioned before, we model only one step of compression F which selects q elements from q^2 (see section 3.2). We set $q = 19$ and use only $q^2 = 361$ first training examples during the compression in the set A . Compression is done by method from section 3.2 with family of tests described in the section 4.2.

4.4 Empirical Measure of Representativeness

Suppose $U = \{z_1, \dots, z_q\}$ is a subset of the training set Z and I is a number of a subset of attributes. Let M_I be leave-one-out 1-nearest-neighbour regression restricted to the attributes from I -th set, i.e.,

$$\alpha_i = |y_i - (y_j \mid j \neq i, \text{dist}^I(x_i, x_j) \rightarrow \min)|. \quad (11)$$

The quality of the prediction of M_I on U is the mean value of $\alpha_1, \dots, \alpha_q$. Let us denote it as $Q(I, U)$. Let us choose the minimal one:

$$I_U = \arg \min_I \{Q(I, U)\}. \quad (12)$$

(If the minimum is reached several times, the minimal I is preferred.)

So I_U is a number of a set of features such that leave-one-out prediction on U is the best when the attributes are restricted to I_U -th set. Check now how I_U fits the whole dataset.

Suppose $Z = \{z_1, \dots, z_n\}$ is the whole training set and $Z_T = \{z_{n+1}, \dots, z_{n+k}\}$ is the testing set. For each of the testing examples, consider

$$\alpha_{n+j} = \min_{i=1}^n \text{dist}^I(z_i, z_{n+j}). \quad (13)$$

Let $Q(I, Z, Z_T)$ be the mean value of $\alpha_{n+1}, \dots, \alpha_{n+k}$.

The overall quality corresponding to U is defined as

$$Q(U) = Q(I(U), Z, Z_T). \quad (14)$$

If $Q(U)$ is small then U is better as a representative subset of Z .

4.5 Experimental Results

Let us use $q^2 + q$ subsets defined in the section [4.2](#)

$$A_1, \dots, A_q; A_{1,1}, \dots, A_{q,q} \quad (15)$$

as a 'control group' for the selection $B = F(Z)$ obtained by our method.

We compare $Q_0 = Q(B)$ with $Q_1 = Q(A_1), \dots, Q_{q^2+q} = Q(A_{q,q})$. The measure of success is the percentage of Q_i less than Q_0 .

In the described experiment it is 95%, or 361 of 380. More concretely, $Q(B) = 2.46$, while the whole range of Q_i is from 2.25 to 7.52. The median value of Q_i is 3.08. If the whole training data is used for selection of attributes, $Q(Z) = 2.54 > Q(B)$.

What does this mean? The size of B is the same as the size of A_i . Subsets A_i are practically random selections of the size q from z_1, \dots, z_{q^2} . They have about 'normal' (i.e. uniformly distributed) level of representativeness with mean 'success' 50% (in the sense mentioned above). As success of S is close to 100%, this means that S is much more representative than a random selection.

5 Conclusion

Practically, basic step F_r is being performed $q^2 + q$ times. Calculating $Q(Z)$ with $|Z| \approx q^2$ once takes approximately same amount of time as calculating $Q(A_i)$ with $|A_i| = q$ for q^2 times.

On the other hand, the total randomness deficiency

$$\mu_f(z|A) = \sum_{I=1}^N (-\log f(I, A, z)) \quad (16)$$

could be replaced by approximation, if we only sum $N' < N$ randomly chosen tests $I = i_1, \dots, i_{N'}$ instead of the whole family $I = 1, \dots, N$.

For this database it $Q(B) = 2.44$ instead of 2.46 when $N' = \lceil N/10 \rceil$ is used. This is the main source of time economy.

References

1. Gammerman, A., Vapnik, V., Vovk, V.: Learning by transduction. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp. 148–156. Morgan Kaufmann, San Francisco (1998)
2. Nourtdinov, I., Melluish, T., Vovk, V.: Ridge Regression Confidence Machine. In: Proceedings of the 18th International Conference on Machine Learning (2001)
3. Saunders, C., Gammerman, A., Vovk, V.: Transduction with confidence and credibility. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence, pp. 722–726 (1999)
4. Vovk, V., Gammerman, A., Saunders, C.: Machine-learning applications of algorithmic randomness. In: Bousquet, O., von Luxburg, U., Rätsch, G. (eds.) Advanced Lectures on Machine Learning. LNCS (LNAI), vol. 3176, pp. 444–453. Springer, Heidelberg (2004)

Concepts for Novelty Detection and Handling Based on a Case-Based Reasoning Process Scheme

Petra Perner

Institute of Computer Vision and applied Computer Sciences,
IBaI Arno-Nitzsche-Str. 43, 04277 Leipzig
pperner@ibai-institut.de
www.ibai-research.de

Abstract. Novelty detection, the ability to identify new or unknown situations that were never experienced before, is useful for intelligent systems aspiring to operate in environments where data are acquired incrementally. This characteristic is common to numerous problems in medical diagnosis and visual perception. We propose to see novelty detection as a case-based reasoning process. Our novelty-detection method is able to detect the novel situation, as well as to use the novel events for immediate reasoning. To ensure this capacity we combine statistical and similarity inference and learning. This view of CBR takes into account the properties of data, such as the uncertainty, and the underlying concepts, such as storage, learning, retrieval and indexing can be formalized and performed efficiently.

1 Introduction

Novelty detection, recognizing that an input differs in some respect from previous inputs, can be a useful ability for learning systems.

Novelty detection is particularly useful where an important class is under-represented in the data, so that a classifier cannot be trained to reliably recognize that class. This characteristic is common to numerous problems, such as information management [1], medical diagnosis [2], fault monitoring and detection [3], and visual perception [4].

In medical image diagnosis, there may be digital images of different modalities showing visual patterns that are referring to a particular disease or, in a simpler case, the interpretation result of such an image just gives a symptom for further medical reasoning.

A prominent application is cell-image analysis. Cell-based assays are used for different purposes: either for diagnostic purposes or for drug development. Hep-2 cell image interpretation is one example for diagnostic purposes. HEp-2 cells are used for the identification of antinuclear autoantibodies (ANA). ANA testing for the assessment of systemic and organ-specific autoimmune diseases has increased progressively since immunofluorescence techniques have first been used to demonstrate antinuclear antibodies in 1957. Hep-2 cells allow for recognition of over 30 different nuclear and cytoplasmic patterns, which are given by upwards of 100 different autoantibodies [5].

Treatment changes, the aging of the population and other medical factors, may change the visual appearance of a pattern, or even a new pattern may appear. The first issue relates to concept drift, whereas the second issue is a novelty-detection problem. An automatic image-diagnosis system should be able to detect the new pattern as a novel event and should also allow to include this novel pattern into the system for further reasoning. Therefore, the system has to analyze the images for the objects-of-interest, then to calculate image features from the discovered objects and, finally, the new pattern must be checked against the existing pattern, based on the calculated image features. When the pattern does not belong to one of the existing classes, the pattern is recognized as a novel event. The novel pattern is introduced into the system by calculating the right attributes and their relevance and by updating the detector based on information about the novel event.

Our novelty-detection approach we describe in this paper is strongly linked to the Case-Based Reasoning (CBR) methodology. That means that we have to treat our novelty detection as a CBR problem. The CBR-based novelty detection consists in successively evolving the previously obtained solutions, taking the data properties, the user's needs and any other prior knowledge into account.

In this paper we propose a general framework for novelty detection based on the CBR methodology [6]. We have developed different concept for novelty detection based on CBR[13]. We studied on a conceptual level how they can be applied to medical problems. One of these concepts is described in this paper. It uses a combination of statistical and similarity-based methods as a solution to the problems underlying the CBR methodology. Our scheme is different from existing work [7]-[10] on novelty detection in that respect that it can perform novelty detection and handling, and it considers the incremental nature of the data.

In Section 2, we describe our proposal for novelty detection and handling. The decision criterion for novelty detection is described in Section 3. Novelty event handling based on similarity inference and case-base management is described in Section 4. The statistical learning method for up-dating existing models and the learning of new models is described in Section 5. In Section 6 we discuss the evaluation issues of the proposed approach. In Section 7 we give a summary and conclusions are given in Section 8.

2 New Proposal for Novelty Detection and Handling

We propose novelty detection to be seen as a case-based reasoning problem [6]. According to our understanding of the novelty detection problem, the case-based reasoning process, with its different tasks, has all the functions necessary for handling novelty detection in an efficient way and it satisfies the incremental nature that it is up to many real-world problems. CBR solves problems using the already stored knowledge, and captures new knowledge, making it immediately available for solving the next problem. Therefore, case-based reasoning can be seen as a method for problem solving, and also as a method to capture new experience and make it immediately available for problem solving. It can be seen as a learning and knowledge-discovery approach, since it can capture from new experience some general knowledge, such as case classes, prototypes and some higher-level concept. We take case-based reasoning

as the framework to solve our novelty detection problem under which we can run the different theoretical methods that should be used to detect the novel events and handle them.

We chose a scenario for our study for which an attribute-value based representation is suitable. Nonetheless, the general framework we propose for novelty detection can be based on any representation. However, the theoretical methods used to solve the different tasks might then be different to the ones we propose here. Thus the representation used to describe our events is an n -dimensional feature vector. This n -dimensional vector should contain as many features as possible for the description of the events collected so far. That makes our problem to a high-dimensional problem. To ensure sufficient performance of our reasoning process, we have to reduce the dimensionality of our problem. Therefore, the feature-selection unit selects from the whole set of features those features that are relevant to describe the known events. Feature selection is based on the conceptual merit algorithm [24].

The heart of our novelty detector (see Fig. 1) is a set of statistical models that have been learnt in an off-line phase from a set of observations. Each model represents a case-class. The probability-density function implicitly represents the data and prevents us from storing all the cases of a known case-class. This unit acts as a novelty-event detector by using the Bayesian decision-criterion with the mixture model. Since this set of observations might be limited, we consider our model as far from optimal, and update the model based on new observed examples. This is done based on the MML-learning principle. Since updating the model based on single events might badly influence the learnt model [11], and is computationally expensive, we collect a sufficiently large set of samples in a data base before starting to update the model.

Therefore, samples that have been detected as a known event are stored under their class label in a data base. After a certain number of samples have been collected for the particular class, the model will be updated based on the MML-learning principle.

In case our model bank cannot classify an actual event into one of the case-classes, this event is recognized as a novel event. Before this event is given to the similarity-based reasoning unit, it is prescreened for outlier. Therefore, the similarity to the representative of the case-class is determined locally on the attribute values, and globally over all attributes. If there is a big deviation in one attribute value, but the overall similarity gives evidence that the sample might belong to one of the case classes, it is displayed to the user. Based on this information and his domain knowledge, the user will decide to label this event as outlier or novel event. Alternatively, this pre-screening step can be skipped and each novel event will be inserted into the similarity-based reasoning unit. This would make the process automatic, but on the other hand the case base might store too many single events that are not useful. Then a special “forgetting strategy” [12] has to be incorporated into the case-base maintenance process.

The novel event is given to the similarity-based reasoning unit. This unit incorporates this sample into their case base according to a case-selective registration-procedure that allows learning case-classes as well as the similarity between the cases and case-classes. We propose to use a fuzzy similarity measure to model the uncertainty in the image data. By doing so, the unit organizes the novel events in such a fashion that is suitable for learning a new statistical model. In contrast to the statistical model, where the probability-density function summarizes the events belonging to

one case-class, the cases in the case base represent explicit knowledge and sufficient storage capacity is needed to keep them.

The case base maintenance unit interacts with the statistical learning unit and gives an advice when a new model has to be learnt. The advice is based on the observation that a case-class is represented by a sufficiently large number of samples that are most dissimilar to other classes in the case-base.

The statistical learning unit takes this case class and proves, based on the MML-criterion, if it is suitable to learn a new model. In case the statistical component recommends not to learn a new model, the case-class is still hosted by the case-base maintenance unit and further up-dated, based on new observed events that might change the inner-class structure, as long as there is new evidence to learn a statistical model.

The similarity-based reasoning unit and the statistical models also act together on the reasoning level. A new observation is first given to the statistical models. If they cannot recognize the new event as belonging to one of their classes, the similarity-based unit finds out if there is a similar event in their case base. In case a similar event is found, the solution associated with the closest case is given as output, and the event is stored in the case base, based on the case-selective case-registration procedure. This procedure ensures that the off-line learnt classes can be handled for reasoning, as well as the new observed novel-events. In this respect the system is not only a novelty-detector, it is also able to handle the novel events and make them immediately available for further reasoning.

In summary, our case-based reasoning process for novelty detection fulfills the following requirements:

1. learning of a (statistical) model for the normal events, as well as
2. updating the model according to new observations, in order to obtain a better model,
3. learning the importance of features for the model according to the observations,
4. recognizing a new event and make it immediately available for further reasoning, and
5. collecting of as much data for the novel event in a structured and incremental fashion as necessary, in order to change from a weak reasoning approach to a strong (statistical) model approach for the recognition of the new event class.

The use of a combination of statistical reasoning and similarity-based reasoning allows implicit and explicit storage of the samples. It allows further to handle well-represented events, as well as rare events.

The above described process can in addition be extended to the specific needs of a classification process [13]. This should be briefly mentioned here, although it is out of the scope of this paper. We can also imagine that the feature description used so far might not be sufficient after having seen more observations. Therefore, our novelty detector should also have a method that can learn new case descriptions. Since new features and observation might change the relevance of the features, we also consider an incremental feature-selection procedure and prototype updating. This goes along with the life-time of a CBR system, and proper procedures for these two tasks will be developed during our study. Since new features and new feature importance change

the case structure and the model, we also have to take into account the architectural aspects of the system. The system needs to have sufficient storing capacity, as well as a function that allows changing the description of a data base.

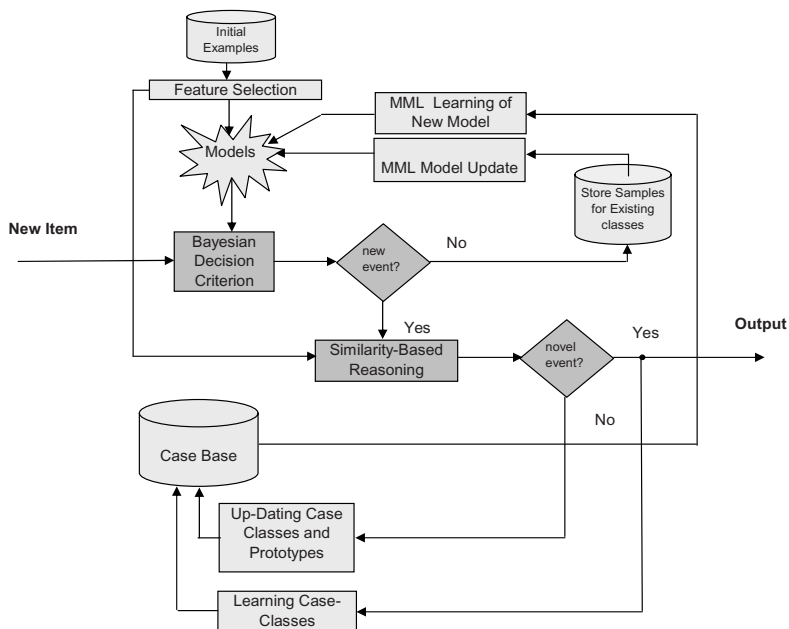


Fig. 1. Our Novelty Detection Schema

3 Bayesian Decision Making for Novelty Detection

When an event occurs, the first step consists of building a detector that allows recognizing if this event is similar to previously occurring events or if it is new. This detector accepts as input the occurring event and the data model and provides as output the decision and the best case class to which the event belongs. In other words, the detector completes the missing data by providing the cluster label. Let us consider x to be an occurring event, D the available data. The event is similar to existing events when the predictive probability-density function is high:

$$p(x/D) = \int p(\theta/D)p(x/\theta)d\theta \quad (1)$$

This Bayesian framework for making predictions can be used for all possible data models. Generally speaking, the integral in this equation is intractable. Several approximation methods exist, such as the Bayesian variational method [14], maximum a posteriori (MAP) method [15]. Following the MAP approximation, we use a single-mind model $\hat{\theta}$ that can maximize the posterior $p(\theta/D)$. By setting the approximate predictive probability distribution:

$$p(x/D) = p(\hat{\theta}/D)p(x/\hat{\theta}) \quad (2)$$

By setting the posterior to unity, assuming that $p(x/D) = \sum_k \alpha_k p(x/\hat{\theta}_k)$ and any x $\max_k p(\theta_k/x)$ belongs to only one cluster. The detector we built consists in maximizing the posterior probability. Alternatively, we can introduce an acceptance level Q requiring that $p(\theta_i/x) \geq Q$. That would prevent us from accepting events with low posterior probability that might have some advantages for the data-storage strategy of the case base, but the distribution is a convolution with a window function. This is a Bayesian decision rule [14]. The retrieval criterion is implemented by using the Bayesian decision rule.

The data model built incrementally can be either the Gaussian mixture, or the mixture of Dirichlet distributions [16]. Traditional parametric inference considers models that can be indexed by a finite-dimensional parameter, for example, the mean and covariance matrix of a multivariate normal distribution of the appropriate dimension. In many cases, however, constraining inference to a specific parametric form, may limit the scope and type of inferences that can be drawn from such models [17]. The Dirichlet distribution has a highly flexible shape, and it is suitable for modeling symmetrical and asymmetrical data. However, in the case of an asymmetrical shape, we are not able to introduce an acceptance level. Therefore, we prefer to use a Gaussian mixture.

4 Novelty Event Handling and Case-Based Maintenance

Once a new event has been inserted into the case base, it should be made immediately available for reasoning. The novel events might be rare, or it might take some time to collect a sufficiently large set of samples for one case class, in order to be able to learn the pdf for the statistical model. Therefore, the choice is to use similarity-based reasoning and to collect the new events in a hierarchical fashion, based on a similarity relation, into the case base. Similar cases representing one new event should be grouped together in a case class, and new events making up a new case class should be inserted into the case base by taking into account the similarity relation to other case classes.

Therefore, the similarity-based reasoning unit is comprised of two functions: 1. reasoning over novel events and 2. collecting new events.

4.1 Similarity-Based Reasoning

For the similarity-based reasoning approach we need an evaluation function that gives us a measure for the similarity between two cases. The chosen similarity-based approach should satisfy the needs for the MML-based learning of the statistical models. The cases should cover the solution space in such a way that it is possible to approximate the final distribution for the statistical models. Therefore we allow the case classes to overlap. The case classes are learnt on a fuzzy conceptual clustering approach. For the similarity-based reasoning we use a fuzzy similarity measure [18].

Let x_i be the new case and m_k ($k = 1, \dots, M$) be k th case class. The distance between the new case x_i and the k th case class should be the minimum value of the fuzzy objective function

$$d(x_i, m) = \min_u \sum_{k=1}^M u_{kt}^n d(x_i, m_k) \quad (3)$$

where $n > 1$ is the degree of fuzziness, u_{kt} is the fuzzy membership function with the case x_i for the k th case classes and satisfies

$$0 \leq u_{kt} \leq 1 \quad \sum_{k=1}^M u_{kt} = 1 \quad (4)$$

The Fuzzy objective function is minimized when

$$u_{kt} = \left[\sum_{l=1}^M [d(x_i, m_k) / d(x_i, m_l)]^{\frac{1}{n-1}} \right]^{-1} \quad (5)$$

Replacing (5) in (3) gives the minimum of the fuzzy objective function

$$d(x_i, m_k) = \left[\sum_{k=1}^M [d(x_i, m_k)]^{\frac{1}{1-n}} \right]^{1-n} \quad (6)$$

The selection criteria would be

$$\text{Select Case } x_i \text{ IF } u_j(x_i) > u_i(x_i) \text{ for all } j \neq i \quad (7)$$

If we have a hierarchy of case classes, this comparison is then done on each level, until we reach a final node.

4.2 Learning the Organization of the Case Base

The aim of CBR learning is to group cases together into groups of similar cases in the case base. The case base should be organized in a hierarchical fashion. More general case groups are located at the top of the hierarchy and more specific case groups can be found when tracing down the hierarchy. That allows efficiently retrieving cases for similarity-based reasoning and gives us a scheme for the collection of a sufficiently large number of similar cases for learning new statistical models. The algorithm is of incremental fashion. That satisfies our incremental collection of cases.

The algorithm [19][20] incrementally incorporates cases into the classification tree, where each node is a prototypical concept that represents a case class. During the construction of the classification tree the new item gets tentatively classified through the existing classification tree. Thereby different possibilities are tried for placing an observation function into the hierarchy:

1. The object is placed into an existing case class,
2. A new case class is created,
3. Two existing case classes are combined into a single case class and
4. an existing node is split into two new case classes.

Depending on the value of the utility function for each of the four possibilities, the observation function gets finally placed into one of these four possible places.

The scoring function for learning this hierarchy is based on the fuzzy intra-class and inter-class variance:

$$Score = \frac{1}{M} \left(\frac{1}{M} \sum_{k=1}^M d^2(m_k, \bar{x}) - \left(\sum_{k=1}^M \frac{1}{\pi_k} \sum_{t=1}^N u_{kt}^n d^2(x_{kt}, m_k) \right) \right) \quad \text{with} \quad \pi_k = \sum_{t=1}^N u_{kt} \quad (8)$$

The normalization to M allows comparing possible different cluster numbers.

The organization of the case base also allows us to take care of rare events. As long as case groups do not give a sufficiently large enough case class, they are represented by higher case groups. If case classes are not used within a certain time period, they can be deleted from the case base.

To our knowledge there exists no algorithm for Fuzzy conceptual clustering yet.

5 MML-Based Learning of the Statistical Models

The Minimum Message Length Principle (MML) [21] can be used as a learning approach for updating the existing statistical model, as well as for learning new statistical models, when enough data are available within the temporary collection. From an information-theory point of view, the minimum message length approach is based on evaluating statistical models according to their ability to compress a message containing the data. High compression is obtained by forming good models of the data to be coded. For each model in the model space, the message includes two parts. The first part encodes the model, using only prior information about the model and no information about the data. The second part encodes only the data, in a way that makes use of the model encoded in the first part [22]. According to information theory [22], the optimal number of clusters of the mixture is that which requires a minimum amount of information to transmit the data efficiently from a sender to a receiver. The message length is defined as minus the logarithm of the posterior probability [22].

It has been shown that when this Bayesian information theory criterion is used with the Gaussian mixture of pdfs, it performs several established criteria of model selection. To the best of our knowledge the MML has not been used before for novelty detection. Let us consider a set of available data $X = (X_1, \dots, X_N)$ controlled by a mixture of M distributions with the parameters $\theta = (\theta_1, \dots, \theta_M)$, and where θ_k is a vector which contains the parameters of the k th distribution. The new data to be added to the existing model is $Y = (Y_1, \dots, Y_L)$. The whole amount of data is $D = (X, Y_L)$. Let us consider that the updating consists in adding a subset of new L_e samples to existing classes, and that the remaining subset with L_n samples allows to create a new class; $L = L_e + L_n$. $L_e = 0$ means that all samples are added to a new class and any change in the data model of existing classes is a consequence of adding this new class. When $L_n = 0$, it means that there is no class creation, and the change in the model data is due to adding new samples in the existing classes. More roughly speaking, after updating, if $L_n = 0$, then the number of classes is $M_n = M$. The new data

model $\tilde{\Theta}$ is obtained by updating the previous model Θ , when new data is added to existing classes. In other words, Θ is equal to θ obtained during the previous update. However, if $L_n \neq 0$, then the number of classes is $M_n = M + 1$. The new data model $\tilde{\Theta}$ is obtained by updating the previous model Θ , when new data is added to a new class and eventually to existing ones. $\Theta = (\theta, \theta_{M+1})$, where θ is the data model obtained during the previous update and modified to fulfill any additional constraints due to the adding of a new cluster. For example, in the case of a mixture model, adding a new cluster implies that the summation of mixing parameters must be normalized such that it is equal to one. θ_{M+1} is the data model of the new class, computed by using a moment method or any other available method.

In the following we will focus on the MML. The MML can be used to check if the new data Y is new or not. For instance, if the MML of the data model, when adding a new cluster, is less than the MML, when these data are added to an existing cluster, then we can decide that the cluster is new; if $MML(M+1) < MML(M)$, then a new cluster is created. The naive computation of this decision rule requires taking all the data $D = (X, Y_L)$ into account, which is time-consuming and therefore limits the usefulness of this decision rule. To overcome this drawback, we propose to write the MML in recursion form; that is the MML_b of the whole data is the MML_a of the data available, plus additional terms related to the new data. The formula is given at the end of the paper.

Suppose now that a new data \vec{X}_{N+1} is inserted in a case. The problem now is how to update the different mixture model parameters. For this goal, we use the stochastic ascent gradient parameter updating proposed in [26]:

$$\Theta^{N+1} = \Theta^N + \gamma_N \left(\frac{\partial(p(\vec{X}_{N+1}, \vec{Z}_{N+1} / \Theta^N))}{\partial \Theta} \right) \quad (9)$$

where \vec{Z}_{N+1} is the missed data vector, γ_N is a sequence of positive numbers.

6 Evaluation Issues for the Proposed Method

A typical medical-image application for novelty detection is the application described in [23]. The image data set is obtained from HEP-2 cell images, comprised of a high number of features, and the class label. These data sets are coming from different manufacturers. The variation among the data is very high since every company uses different imaging and cell lines. Nonetheless, the expected output, the class label, should be the same. That means that the class number in the table represents for each class the same class label. Some classes have a sufficiently large number of samples, while others are under-represented. The sample distribution is shown in table 2.

Only database DB_1 can be used for the learning of a statistical model, since the four classes have a sufficiently large number of samples. Class 1 to 4 in database DB2-DB4 can be used to update the statistical model. All the other classes in the databases should be considered as novel events that occur in sequence.

The evaluation of our method is not easy because of the underlying different aspects. These aspects are:

1. up-dating existing concepts to achieve a better performance of the model or to handle the concept drift,
2. the recognition of novel events,
3. reasoning over novel events and
4. the learning of new concepts.

At the recent status we can only give an outline of the evaluation procedure and the concept behind.

These four tasks might be influenced by different factors:

1. Up-dating existing Concepts
2. There must be some influence of the sample distribution.
3. How many samples are necessary for evaluating the recent performance of the model? According to the literature approximately 100 samples [26] are needed.
4. Recognition of novel events and outliers
5. This has something to do with false/positive recognition
6. Learning New Concepts
7. How many samples are necessary to learn novel concepts? According to the literature [25] approximately 100 samples are needed.
8. How can the case base organization control the learning of the models?

Table 1. Name of Database and Number of Classes and Samples per Class

Name	Class Number																										Number of Classes	Number of Cases
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26		
DB_1	105	96	63	83																							4	347
DB_2	8	2	2	14	7	5	15	9	5	4	14	9	8	10	48	23	17	31	3	3	3	7	5	2	13	5	26	298
DB_3	7	30	29	28	11	7	13	5	13	13																	10	156
DB_4	25	12	18	21	5	16	22	24	21	20	5	14															12	203

The evaluation of the statistical model can only be done by test-and-train based on a large enough test data set (500 samples according to [25]) which is not available in the proposed kind of application. We rather have to do with a sparse-data set problem. Therefore, we can only evaluate the generalization error of the model, that is the error rate we obtain when presenting the data set to the system that has also been used for learning. Besides that we can calculate the error rate we obtain when presenting the samples belonging to the same class 1 to 4 in the data base DB_2, DB_3, and DB_4 to the models.

7 Discussion

The idea to combine statistical reasoning and similarity-based reasoning is based on the fact that statistical reasoning has a long history in handling uncertain and imprecise data and implicitly represents the data. That prevents us from having a large data-storing capacity for the system. However, statistical reasoning is not good in handling

single events and improving reasoning over time when new data arrive. Therefore, we have introduced the MML-based learning approach to update the model. This up-dating step can be seen as an adaptation of the solution to the current case.

Similarity-based reasoning can handle the single event, for reasoning as well as for organizing it into such a fashion that it can be used for up-dating the model. It can also help to detect outliers. However, the similarity-based reasoning unit explicitly represents the data and therefore enough storage capacity has to be provided for the system.

The hierarchical organization of the case base can also control this MML-based learning process by advising the learning unit to learn a coarse model first and to learn a specialized model when enough data is available in the respective subnode of the hierarchical case base.

In general we have to say that the proposed approach is only applicable if we have enough data samples. The kind of medical applications we usually have to deal with do not fulfill this requirement. Only after the system is in use we can collect enough data for statistical modeling. The idea can be to start with a pure similarity-based approach and create initial statistical models off-line after enough data are available for one class.

8 Conclusion

We have outlined in this paper our new approach to novelty detection, which is based on the idea to see novelty detection as a CBR process. Our novelty-detection method is able to detect the novel situation, as well as to handle the novel events for immediate reasoning. We combine statistical and similarity inference and learning. This view of CBR takes into account the properties of data such as the uncertainty, and the underlying concepts such as adaptation, storage, learning, retrieval and indexing can be formalized and performed efficiently.

Known classes are described by statistical models. The performance of the models is improved by the incremental up-dating of the models based on new available events. Since up-dating the model based on single events is not appropriate, a sufficiently large number of cases is collected into a temporary case collection. This case collection is emptied after the model is up-dated. The information about the cases is now implicitly represented by the model, and storage capacity is preserved. New events, not belonging to one of the known case classes, are recognized as novel events. These events are stored by a similarity-based registration procedure in a second case base. The similarity-based learning procedure ensures that similar cases are grouped into case classes, a representative for the case class is learnt and generalization over case classes can be performed. This allows one to efficiently collect novel events and group them in such a way that retrieval over the case base is efficient. The similarity-based unit is also responsible for making novel events immediately available for reasoning.

When a sufficiently large number of cases for a case class is available in the second case base, the case class is given to the statistical learning unit for learning a new statistical model. The statistical learning strategy for up-dating a model and learning

new models is based on the MML principle. Now the new case class is handled further by the statistical model and entry in the second case base can be deleted.

Acknowledgement. This work has been supported by the German Ministry of Science and Technology BMBF under the grant-No. CAN 06/A07. The author would like to thank Michael Richter and Ron Kenett for their advice and helpful comments.

References

- [1] Schiffmann, B., McKeown, K.R.: Context and Learning in Novelty Detection. In: Proc. HLT-EMNLP 2005, Vancouver, BC (October 2005)
- [2] Spinosa, E.J.: André Carlos Ponce Leon Ferreira de Carvalho: SVMs for novel class detection in Bioinformatics. WOB 2004, 81–88 (2004)
- [3] Liang, B., Austin, J.: Mining Large Engineering Data Sets on the Grid Using AURA. In: Yang, Z.R., Yin, H., Everson, R.M. (eds.) IDEAL 2004. LNCS, vol. 3177, pp. 430–436. Springer, Heidelberg (2004)
- [4] Singh, S., Markou, M.: An approach to novelty detection applied to the classification of image regions. *IEEE Transactions on Knowledge and Data Engineering* 16(4), 396–407 (2004)
- [5] Bradwell, A.R., Stokes, R.P., Johnson, G.D.: Atlas of HEP-2 Patterns. AR Bradwell (1995)
- [6] Althoff, K.D.: Case-Based Reasoning. In: Chang, S.K. (ed.) Handbook on Software Engineering and Knowledge Engineering (2001)
- [7] Markow, M., Singh, S.: Novelty Detection: A Review-Part 1: Statistical Approaches. *Signal Processing* 83(12), 2481–2497 (2003)
- [8] Markow, M., Singh, S.: Novelty Detection: A Review-Part 2: Neural Network Based Approaches. *Signal Processing* 83(12), 2499–2521 (2003)
- [9] Zhang, Y., Callan, J., Minka, T.: Novelty and Redundancy Detection in Adaptive Filtering. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 81–88. ACM Press, New York (2002)
- [10] Tax, D.M.J., Jusczyk, P.: Kernel Whitening for One-Class Classification *International Journal of Pattern Recognition and Artificial Intelligence* (2003)
- [11] Bishop, Ch.M.: Pattern Recognition and Machine Learning. LNCS. Springer, Heidelberg (2006)
- [12] Leake, D.B., Wilson, D.C.: Remembering why to remember: performance-guided case-base maintenance. In: Blanzieri, E., Portinale, L. (eds.) *Advances in Case-Based Reasoning*, pp. 161–172. Springer, Heidelberg (2000)
- [13] Perner, P.: Concepts for Novelty Detection and Handling based on Case-Based Reasoning, IBAI-Report (October 2006)
- [14] Berger, J.: *Statistical Decision Theory and Bayesian Analysis*. LNCS. Springer, Heidelberg (1985)
- [15] MacKay, D.J.C.: *Information Theory, Inference and Learning Algorithm*. Cambridge University Press, Cambridge (2003)
- [16] Kotz, S., Ng, K.W., Fankg, K.: *Symmetric Multivariate and Related Distributions*. Chapman and Hall, London / New York (1990)
- [17] Sjolandery, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., Saira Mian, I., Haussler, D.: Dirichlet Mixtures: A Method for Improved Detection of Weak but Significant Protein Sequence Homology. *Computer Applications in the Biosciences* (1996)

- [18] Bezdek, J.C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA (1981)
- [19] Jaenichen, S., Perner, P.: Conceptual Clustering and Case Generalization of two dimensional Forms. *Computational Intelligence* 22(3/4), 177–193 (2006)
- [20] Perner, P.: Case-base maintenance by conceptual clustering of graphs. *Engineering Applications of Artificial Intelligence* 19(4), 295–381 (2006)
- [21] Wallace, C.S.: *Statistical and Inductive Inference by Minimum Message Length*. Information Science and Statistics. Springer, Heidelberg (2005)
- [22] MacKay, D.J.C.: *Information Theory, Inference and Learning Algorithm*. Cambridge University Press, Cambridge (2003)
- [23] Perner, P.: *Prototype-Based Classification*. *Applied Intelligence* (to appear)
- [24] Hong, S.J.: Use of contextual information for feature ranking and discretization. *IEEE Trans. on Knowledge Discovery and Data Engineering*, 55–65
- [25] Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of Finite-Mixture Models. *IEEE Trans. on PAMI* 24(3), 381–396
- [26] Gill, P.E., Murray, W., Wright, M.H.: *Practical Optimization*. Academic Press, San Diego (1981)

An Efficient Algorithm for Instance-Based Learning on Data Streams

Jürgen Beringer¹ and Eyke Hüllermeier²

¹ Fakultät für Informatik

Otto-von-Guericke-Universität Magdeburg

beringer@iti.cs.uni-magdeburg.de

² Fachbereich Mathematik und Informatik

Philipps-Universität Marburg

eyke@mathematik.uni-marburg.de

Abstract. The processing of data streams in general and the mining of such streams in particular have recently attracted considerable attention in various research fields. A key problem in stream mining is to extend existing machine learning and data mining methods so as to meet the increased requirements imposed by the data stream scenario, including the ability to analyze incoming data in an online, incremental manner, to observe tight time and memory constraints, and to appropriately respond to changes of the data characteristics and underlying distributions, amongst others. This paper considers the problem of classification on data streams and develops an instance-based learning algorithm for that purpose. The experimental studies presented in the paper suggest that this algorithm has a number of desirable properties that are not, at least not as a whole, shared by currently existing alternatives. Notably, our method is very flexible and thus able to adapt to an evolving environment quickly, a point of utmost importance in the data stream context. At the same time, the algorithm is relatively robust and thus applicable to streams with different characteristics.

1 Introduction

In recent years, so-called *data streams* have attracted considerable attention in different fields of computer science. As the notion suggests, a data stream can roughly be thought of as an ordered sequence of data items, where the input arrives more or less continuously as time progresses [16]. There are various applications in which streams of this type are produced, such as network monitoring or telecommunication systems.

Apart from other issues such as data processing and querying, the problem of mining data streams has been studied in a number of recent publications (see e.g. [13] for an up-to-date overview). In this connection, different data mining problems have already been considered, such as clustering [1], classification [17], and frequent pattern mining [8]. In this paper, we are concerned with the classification problem. More specifically, we investigate the potential of the instance-based

approach to supervised learning within the context of data streams and propose an efficient instance-based learning algorithm.

The remainder of the paper is organized as follows: Section 2 provides some background information, both on data streams and on instance-based learning, and briefly reviews related work. Our approach to instance-based learning on data streams is introduced in section 3 and empirically evaluated in section 4. The paper concludes with a brief summary in section 5.

2 Background

2.1 Data Streams and Concept Change

The *data stream model* assumes that input data are not available for random access from disk or memory, such as relations in standard relational databases, but rather arrive in the form of one or more continuous data streams. The stream model differs from the standard relational model in various ways [4]: (i) The elements of a stream arrive incrementally in an “online” manner. That is, the stream is “active” in the sense that the incoming items trigger operations on the data rather than being sent on request. (ii) The order in which elements of a stream arrive are not under the control of the system. (iii) Data streams are potentially of unbounded size. (iv) Data stream elements that have been processed are either discarded or archived. They cannot be retrieved easily unless being stored in memory, which is typically small relative to the size of the stream (stored/condensed information about past data is often referred to as a *synopsis*). (v) Due to limited (memory) resources and strict time constraints, the computation of exact results will often not be possible. Therefore, the processing of stream data does commonly produce *approximate* results.

For the problem of mining data streams, the aforementioned characteristics have a number of important implications. First of all, in order to guarantee that results are always up-to-date, it is necessary to analyze the incoming data in an online manner, tolerating not more than a constant time delay. Since learning from scratch every time is generally excluded due to limited time and memory resources, corresponding learning algorithms must be *incremental*. Moreover, algorithms for learning on data streams must also be *adaptive*, i.e., they must be able to adapt to an evolving environment in which the data (stream) generating process may change over time. Thus, the handling of changing concepts is of utmost importance in mining data streams [5]. It has not only been considered in this context, however. In general, the literature distinguishes between different types of concept change over time [27]. The first type refers to a sudden, abrupt change of the underlying concept to be learned and is often called *concept shift*. Roughly speaking, in case of a concept shift, any knowledge about the old concept will typically become obsolete and the new concept has to be learned from scratch. The second type refers to a gradual evolution of the concept over time. In this scenario, old data might still be relevant, at least to some extent. Finally, one often speaks about *virtual* concept drift if not the concept itself changes

but the distribution of the underlying data generating process [29]. Note that in practice virtual and real concept drift can occur simultaneously.

Concept change can be handled in a direct or indirect way. In the indirect approach, the learning algorithm does not explicitly attempt to detect a concept drift. Instead, the use of outdated or irrelevant data is avoided from the outset. This is typically accomplished by considering only the most recent data while ignoring older observations, e.g., by sliding a window of fixed size over a data stream or by weighing the nearest neighbors of new observations, not only according to their distance but also according to their age. More generally, such strategies belong to the class of *instance selection* or *instance weighing* methods. To handle concept change in a more direct way, appropriate techniques for discovering the drift or shift are first of all required. Such techniques are typically based on statistical tests. Roughly speaking, the idea is to compare a certain statistic that refers to recently observed data with a corresponding statistic for older data, and to decide whether the difference between them is significant in a statistical sense. (A corresponding technique will be discussed in section 3 below.)

2.2 Instance-Based Learning

As opposed to model-based machine learning methods which induce a general model (theory) from the data and use that model for further reasoning, instance-based learning (IBL) algorithms simply store the data itself. They defer the processing of the data until a prediction (or some other type of query) is actually requested, a property which qualifies them as a *lazy* learning method [3,2]. Predictions are then derived by combining the information provided by the stored examples.

Such a combination is typically accomplished by means of the *nearest neighbor* (NN) estimation principle [9]. Consider the following setting: Let \mathcal{X} denote the instance space, where an instance corresponds to the description x of an object (usually though not necessarily in attribute–value form). \mathcal{X} is endowed with a distance measure $\Delta(\cdot)$, i.e., $\Delta(x, x')$ is the distance between instances $x, x' \in \mathcal{X}$. \mathcal{L} is a set of class labels, and $\langle x, \lambda_x \rangle \in \mathcal{X} \times \mathcal{L}$ is called a labeled instance, a case, or an example. In classification, which is the focus of this paper, \mathcal{L} is a finite (usually small) set comprised of m classes $\{\lambda_1 \dots \lambda_m\}$.

The current experience of the learning system is represented in terms of a set \mathcal{D} of examples $\langle x_i, \lambda_{x_i} \rangle$, $1 \leq i \leq n = |\mathcal{D}|$. From a machine learning point of view, \mathcal{D} plays the role of the *training set* of the learner. More precisely, since not all examples will necessarily be stored by an instance-based learner, \mathcal{D} is only a subset of the training set. In case-based reasoning, it is also referred to as the *case base*; besides, in the context of data streams, \mathcal{D} corresponds to the aforementioned *synopsis*.

Finally, suppose a novel instance $x_0 \in \mathcal{X}$ (a query) to be given, the class label λ_{x_0} of which is to be estimated. The NN principle prescribes to estimate this label by the label of the nearest (most similar) sample instance. The *k-nearest neighbor* (*k*-NN) approach is a slight generalization, which takes the

$k \geq 1$ nearest neighbors of x_0 into account. That is, an estimation $\lambda_{x_0}^{est}$ of λ_{x_0} is derived from the set $\mathcal{N}_k(x_0)$ of the k nearest neighbors of x_0 , usually by means of a *majority vote*:

$$\lambda_{x_0}^{est} = \arg \max_{\lambda \in \mathcal{L}} \text{card}\{x \in \mathcal{N}_k(x_0) \mid \lambda_x = \lambda\}. \quad (1)$$

Regarding the suitability of IBL in the context of data streams, note that IBL algorithms are inherently incremental, since adaptation basically comes down to adding or removing observed cases. On the other hand, this training efficiency comes at the cost of high complexity at classification time, which involves retrieving the query’s nearest neighbors. Consequently, IBL might be preferable (to model-based methods) in a data stream application if the number of incoming data is large compared with the number of queries to be answered, i.e., if model updating is the dominant factor.

2.3 Related Work

Data mining on streams is a topic of active research, and several adaptations of standard statistical and data analysis methods to data streams or related models have been developed recently [12]. Likewise, several online data mining methods have been devised (e.g. [10]), with a particular focus on unsupervised techniques like clustering. Supervised learning on streams, including classification, has received less attention so far, even though some approaches have already been developed.

A very early approach is the FLORA (Floating Rough Approximation) system [30]. The corresponding algorithm learns rule-based binary classifiers on a sliding window of fixed size. The FRANN (Floating Rough Approximation in Neural Networks) algorithm trains RBF networks on a sliding window of adaptive size [22]. The LWF (Locally Weighted Forgetting) algorithm of Salganicoff [25] is among the best adaptive learning algorithms. It is an instance-based learner that reduces the weights of the k nearest neighbors $x_1 \dots x_k$ (in increasing order according to distance) of a new instance x_0 by the factor $\tau + (1 - \tau)\Delta(x_i, x_0)^2 / \Delta(x_k, x_0)^2$. An instance is completely removed if its weight falls below a threshold θ . To fix the size of the case base, the parameter k is adaptively defined by $k = \lceil \beta |\mathcal{D}| \rceil$ where $|\mathcal{D}|$ is the size of the current case base. As an obvious alternative to LWF, Salganicoff considers the TWF (Time Weighted Forgetting) algorithm that weights instances according to their age: at time point t , the example observed at time $t - k$ is weighted by w^k , where $w \in (0, 1)$ is a constant. The Prediction Error Context Switching algorithm (PECS), also proposed in [25], does not delete but only deactivates instances. That is, removed instances are still stored in memory and might be reactivated later on. This strategy can avoid some disadvantages of LWF but entails storage requirements that disqualify PECS for the data stream context.

In the above approaches, the strategies for adapting the size of a sliding window, if any, are mostly of a heuristic nature. In [19], the authors propose to adapt the window size in such a way as to minimize the estimated generalization error

of the learner trained on that window. In [20], this approach is further generalized by allowing for the selection of arbitrary subsets of batches instead of only uninterrupted sequences. Despite the appealing idea of this approach to window (training set) adjustment, the successive testing of different window lengths is computationally expensive and therefore not immediately applicable in a data stream scenario with tight time constraints.

Recently, some efforts have been made to extend decision tree induction to the streaming scenario. In their CVFDT (Continuous Very Fast Decision Trees) algorithm, Hulton and Domingos learn and maintain decision trees on a sliding window of fixed size [17]. Another approach to adaptive learning is the use of ensemble techniques [21]. Here, the idea is to train multiple classifiers, often decision trees, on different blocks of data. To achieve adaptivity, the classifiers are weighted according to their (recent) performance or, even simpler, only the best classifier is selected to classify new instances. If concept drift occurs, outdated or poorly performing classifiers are replaced by new ones.

So-called *editing strategies* for nearest neighbor classification or, more generally, lazy learning have been studied for quite a while [24]. Even though these strategies are of course related to the problem of adaptive learning and handling concept change, they are not suitable for data stream applications, mainly for the following reasons: Firstly, they solely focus on the goal to maximize classification accuracy while disregarding other aspects like space complexity. Secondly, they are not flexible and efficient enough for online classification. In this connection, let us also mention the well-known IB3 algorithm [3], which is built upon IB1 and includes means to delete noisy and old instances that do no longer comply with the current concept. Even though IB3 is thus principally able to handle gradual concept drift, the adaptation is relatively slow [30,25].

Finally, we note that there is also a bunch of work on time series data mining (e.g. [18]). However, even though time series data mining is of course related to stream data mining, one should not overlook important differences between these fields. Particularly, time series are still static objects that can be analyzed offline, whereas the focus in the context of data streams is on dynamic adaptation and online data mining.

3 Instance-Based Learning on Data Streams

This section introduces our approach to instance-based learning on data streams, referred to as IBL-DS. The learning scenario consists of a data stream that permanently produces examples, potentially with a very high arrival rate, and a second stream producing query instances to be classified. The key problem for our learning system is to maintain an implicit concept description in the form of a case base (memory). Before presenting details of IBL-DS in section 3.2, some general aspects and requirements of concept adaptation (case base maintenance) in a streaming context will be discussed in section 3.1.

3.1 Concept Adaptation

The simplest adaptive learners are those using sliding windows of fixed size. Unfortunately, by fixing the number of examples in advance, it is impossible to optimally adapt the size of the case base to the complexity of the concept to be learned, and to react to changes of this concept appropriately. Moreover, being restricted to selecting a subset of successive observations in the form of a window, it is impossible to disregard a portion of observations in the middle (e.g. outliers) while retaining preceding and succeeding blocks of data. To avoid both of the aforementioned drawbacks, non-window-based approaches are needed that do not only adapt the size of the training data but also have the liberty to select an arbitrary *subset* of examples from the data seen so far. Needless to say, such flexibility does not come for free. Apart from higher computational costs, additional problems such as avoiding an unlimited growth of the training set and, more generally, trading off accuracy against efficiency have to be solved.

Instance-based learning seems to be attractive in light of the above requirements, mainly because of its inherently incremental nature and the simplicity of model adaptation. In particular, since in IBL an example has only local influence, the update triggered by a new example can be restricted to a local region around that observation.

Regarding the updating (editing) of the case base in IBL, an example should in principle be retained if it improves the predictive performance (classification accuracy) of the classifier; otherwise, it should better be removed. Unfortunately, this criterion cannot be used directly, since the (future) usefulness of an example in this sense is simply not known. Instead, existing approaches fall back on suitable indicators of usefulness:

- Temporal relevance: According to this indicator, recent observations are considered as potentially more useful and, hence, are preferred to older examples.
- Spatial relevance: The relevance of an example can also depend on its position in the instance space. In IBL, examples can be redundant in the sense that they don't change the nearest neighbor classification of any query. More generally (and less stringently), one might consider a set of examples redundant if they are closely neighbored in the instance space and, hence, have a similar region of influence.
- Consistency: An example might be removed if it seems to be inconsistent with the current concept, e.g., if its own class label differs from those labels in its neighborhood.

Many algorithms use only one indicator, either temporal relevance (e.g. window-based approaches), spatial relevance (e.g. LWF), or consistency (e.g. IB3). A few methods also use a second indicator, e.g. the approach of Klinkenberg (temporal relevance and consistency), but only the window-based system FLORA4 uses all three aspects.

3.2 IBL-DS

In this section, we describe the main ideas of IBL-DS, our approach to IBL on data streams, that not only takes all of the aforementioned three indicators into account but also meets the efficiency requirements of the data stream setting.

IBL-DS optimizes the composition and size of the case base autonomously. On arrival of a new example $\langle x_0, \lambda_{x_0} \rangle$, this example is first added to the case base. Moreover, it is checked whether other examples might be removed, either since they have become redundant or since they are outliers (noisy data). To this end, a set C of examples within a neighborhood of x_0 are considered as candidates. This neighborhood is given by the k_{cand} nearest neighbors of x_0 , and the candidate set C consists of the 50% oldest examples within that neighborhood. The 50% most recent examples are excluded from removal due to the difficulty to distinguish potentially noisy data from the beginning of a concept change. Even though unexpected observations will be made in both cases, noise and concept change, these observations should be removed only in the former but not in the latter case.

If the current class λ_{x_0} is the most frequent one within a larger test environment of size $k_{test} = (k_{cand})^2 + k_{cand}$, those candidates in C are removed that have a different class label. Furthermore, to guarantee an upper bound on the size of the case base, the oldest element of the similarity environment is deleted, regardless of its class, whenever the upper bound would be exceeded by adding the new example.

Using this strategy, the algorithm is able to adapt to concept drift but will also have a high accuracy for non-drifting data streams. Still, these two situations – drifting and stable concept – are to some extent conflicting with regard to the size of the case base: If the concept to be learned is stable, classification accuracy will increase with the size of the case base. On the other hand, a large case base turns out to be disadvantageous in situations where concept drift occurs, and even more in the case of concept shift. In fact, the larger the case base is, the more outdated examples will have to be removed and, hence, the more sluggish the adaptation process will be.

For this reason, we try to detect an abrupt change of the concept using a statistical test as in [14,15]. If a corresponding change has been detected, a large number of examples will be removed instantaneously from the case base. The test is performed as follows: We maintain the prediction error p and standard deviation $s = \sqrt{\frac{p(1-p)}{100}}$ for the last 100 training instances. Let p_{min} denote the smallest among these errors and s_{min} the associated standard deviation. A change is detected if the current value of p is significantly higher than p_{min} . Here, statistical significance is tested using a standard (one-sided) z-test, i.e., the condition to be tested is $p + s > p_{min} + z_\alpha s_{min}$, where α is the level of confidence (we use $\alpha = 0.999$).

¹ This choice of k_{test} aims at including in the test environment the similarity environments of all examples in the similarity environment of x_0 ; of course, it does not guarantee to do so.

Finally, in case a change has been detected, we try to estimate its extent in order to determine the number of examples that need to be removed. More specifically, we delete p_{dif} percent of the current examples, where p_{dif} is the difference between p_{min} and the classification error for the last 20 instances. Examples to be removed are chosen at random according to a distribution which is spatially uniform but temporally skewed (see below).

IBL-DS is implemented under the data mining library WEKA [31]. The data is stored in the M-tree data structure of XXL, a query processing library developed and maintained at the Informatics Institute of Marburg University [6]. Below, we describe the distance function employed by IBL-DS and the M-Tree [7] which allows for processing streams with both continuous and categorical attributes and, moreover, to perform nearest neighbor queries in an efficient way even for very large case memories.

As a distance function we use an updateable variant of SVDM which is a simplified version of the VDM distance measure [26] and was successfully used in the classification algorithm RISE [11]. Let an instance x be specified in terms of ℓ features $F_1 \dots F_\ell$, i.e., as a vector $x = (f_1 \dots f_\ell) \in D_1 \times \dots \times D_\ell$. Numerical features F_i with domain $D_i = \mathbb{R}$ are first normalized by the mapping $f_i \mapsto f_i / (\max - \min)$, where \max and \min denote, respectively, the largest and smallest value for F_i observed so far; these values are permanently updated. Then, $\delta_i(f_i, f'_i)$ is defined by the distance between the normalized values of f_i and f'_i . For a discrete attribute F_j , the distance between two values f_j and f'_j is defined by the following measure:

$$\delta_i(f_j, f'_j) = \sum_{k=1}^m \|P(\lambda_k | F_j = f_j) - P(\lambda_k | F_j = f'_j)\|,$$

where m is the number of classes and $P(\lambda | F = f)$ is the probability of the class λ given the value f for attribute F . Finally, the distance between two instances x and x' is given by the mean squared distance

$$\Delta(x, x') = \frac{1}{\ell} \sum_{i=1}^{\ell} \delta_i(f_i, f'_i)^2$$

To delete instances in a spatially uniform but temporally skewed way, we exploit the properties of the M-Tree index structure [7]. In this tree, the leaves store instances that belong to a small sphere within the instance space. The inner nodes combine subnodes to bigger spheres and the root node represents the sphere that corresponds to the whole data set. Each node n consists of a center instance c_n and an associated radius r_n . Moreover, each node maintains a list of successors (subnodes) l_n . The number of instances or subnodes of a node is restricted to an interval $[\minCapacity, \maxCapacity]$. Our experience has shown that the interval $[6, 15]$ yields good performances. The stored examples correspond to the instance nodes of the tree (located directly under the leaf nodes), the radius of which is 0.

To delete data with preference to older instances, the number of items to be removed in a node is uniformly spread among the subnodes. In a leaf, only the

oldest instances are removed. This way, we ensure that the spatial distribution of the deleted instances is uniform in the instance space. Regarding the temporal distribution, however, old instances are more likely to be removed than more recent examples.

Finally, in order to classify a new query instance x_0 , we employ a simple majority voting procedure among the k nearest neighbors. As in standard IBL, the computationally most expensive step consists of finding the query’s neighbors. In our implementation, this step is again supported by the aforementioned M-tree (more specifically, the nearest neighbors are computed in an iterative way using a (min-)heap H of nodes which is initialized with the root of the M-tree).

4 Empirical Evaluation

A convincing experimental validation of online learning algorithms is an intricate problem for several reasons. Firstly, the evaluation of algorithms in a streaming context is obviously more difficult than the evaluation for static data sets, mainly because simple, one-dimensional performance measures such as classification accuracy will now vary over time and, hence, turn into functions (of time) which are not immediately comparable. Besides, additional criteria become relevant, such as the handling of concept drift, many of which are rather vague and hard to quantify. Secondly, real-world and benchmark streaming data is currently not available in a form that is suitable for conducting systematic experiments.

Due to these reasons, we mainly used synthetic data for our experiments, which allows for conducting experiments in a *controlled* way. Besides, a further experimental study using real-world data is presented in section 4.3.

We compared IBL-DS with the following instance-based approaches: The simple sliding window approach with fixed window-sizes of 200, 400, 800, respectively (Win200, Win400, Win800); Local Weighed Forgetting with $\beta = 0.04$ and $\beta = 0.02$ (LWF04, LWF02); Time Weighed Forgetting with $w = 0.996$ and $w = 0.998$ (TWF996, TWF998).

For IBL-DS we used the parameters $k_{cand} = 5$ and a maximal size of 5,000 examples for the case memory. For nearest neighbor classification, the neighborhood size was set to $k = 5$ for all algorithms.² In order to show the flexibility of IBL-DS, we employed synthetic data with quite different characteristics (see below).

4.1 Performance Measures and Data Sets

The learning scenario we considered is a straightforward extension of supervised learning to the data stream setting: At each point of time a new instance x_0 arrives (from the query stream) and its class label λ_{x_0} is predicted. After the prediction has been made, the correct label is provided by a teacher, the

² Note that the primary purpose of our studies is to compare the algorithms under equal conditions. This is why we used a fixed neighborhood size instead of optimizing this parameter.

prediction is evaluated, and the case base is updated (i.e., the new example is submitted to the example stream).

All data streams were tested with 50,000 elements, using an initial training set of 100 examples and adding 5% random noise. We derived two types of classification rate: (i) The *streaming* classification rate measures the accuracy on the last 100 instances of the stream. Thus, it is a kind of real accuracy that refers to a certain section of the stream. (ii) The *absolute* classification rate aims at estimating the accuracy at a particular moment of time. To this end, 1,000 extra test instances are generated at random according to a uniform distribution, and the classification accuracy for this test set is derived by using the current case base; this is done for every 10 time points.

We conducted experiments with 8 different data streams (see table [II](#)), some of which have already been used in the literature before: The streams GAUSS, SINE2, STAGGER and MIXED were used in [\[14\]](#), and the HYPERPLANE data (that we generated for the dimensions $d = 2$ and $d = 5$) was used in multiple experiments for data streams in [\[28\]](#). Besides, we used the following data streams:

DISTRIB: Instances are uniformly distributed in the unit square $[0, 1] \times [0, 1]$. An instance (x, y) belongs to class 1 if $(x, y) \in [0, 0.5] \times [0, 0.5]$ or $(x, y) \in]0.5, 1] \times]0.5, 1]$, otherwise to class 0. Even though the concept remains fixed, the underlying distribution does change: the data is only generated in one quarter of the instance space, changing the quarter clockwise every 2,000 instances.

RANDOM: Instances are uniformly distributed in the unit square $[0, 1] \times [0, 1]$. An instance (x, y) belongs to class 1 with probability p , independently of x and y . Within an interval of 2,000 examples, the probability p increases linearly from 0 to 1, then decreases linearly from 1 to 0 during the next interval of the same length, and so on.

MEANS: The n classes are defined by n center points in $[0, 1]^d$. An instance belongs to the class of the nearest center. Each center moves with a fixed drift for each dimension. If it leaves the unit interval in one dimension, the drift for this dimension is inverted. We have made experiments with $n = 5$ and $d \in \{2, 5\}$. For each dimension, the drift is initialized by a random value in $[-(1/8)^{-3}, (1/8)^{-3}]$.

Table 1. Properties of data streams

	attributes	classes	drift/shift
GAUSS	2 num	2	shift
SINE2	2 num	2	shift
DISTRIB	2 num	2	virtual shift
RANDOM	2 num	2	drift (distr.)
STAGGER	3 discr	2	shift
MIXED	2 num + 2 discr	2	shift
HYPER	2/5 num	2	drift
MEANS	2/5 num	5	drift

4.2 Results

The absolute and streaming classification rates are shown, respectively, in tables 2 and 3. For the data streams GAUSS, SINE2, RANDOM and MIXED, our method IBL-DS shows the best performance regardless of the type of measure; for DISTRIB it performs best in terms of the absolute classification rate. Even if IBL-DS is not the best method for the remaining streams (STAGGER, HYPER, and MEAN), its results are always competitive and close to optimal.

Apparently, IBL-DS performs comparatively well especially for the data streams with concept shift. Thus, our strategy for handling such situations, including a flexible size of the case base, seems to work in practice. In fact, some other methods do obviously have difficulties with abrupt changes of the concept, as suggested by their relatively poor classification rates. Note that concept shift does also occur in STAGGER. Here, however, only 12 different instances exist, so a small case base is always sufficient. In fact, it is not useful to store all examples that support the current concept; this only makes the model less flexible with regard to the next concept shift but does not lead to a higher accuracy.

Table 2. Absolute classification rates

	IBL-DS	LWF02	LWF04	Win200	Win400	Win800	TWF996	TWF998
GAUSS	.843	.805	.837	.834	.804	.734	.750	.693
SINE2	.919	.863	.898	.896	.868	.788	.838	.762
DISTRIB	.948	.913	.888	.504	.508	.514	.500	.505
RANDOM	.723	.706	.718	.712	.704	.674	.655	.625
STAGGER	.956	.806	.806	.978	.962	.917	.916	.908
MIXED	.906	.713	.707	.898	.870	.790	.840	.765
HYPER2	.969	.970	.965	.959	.965	.963	.923	.919
HYPER5	.904	.892	.896	.876	.886	.880	.839	.834
MEANS2	.944	.950	.935	.918	.939	.946	.913	.914
MEANS5	.809	.828	.796	.736	.778	.810	.735	.763

For the DISTRIB data, the extreme differences between absolute and streaming classification rate call for explanation. To understand these differences recall the special distribution of the training data: After a shift of this distribution, it takes 6,000 time steps (instances) until the next instance for the previous quarter will arrive. All window-based approaches will soon forget all the data of this quarter. Only the LWF algorithm stores all the data the whole time. IBL-DS will have the highest accuracies after training instances have been seen in all quarters (viz. after 6,000 instances). Before, LWF performs slightly better, since this algorithm does not delete as many of the 100 examples used for initialization.

The simple window-based algorithm shows a very good performance for the HYPERPLANE and the MEANS data. Again, there is a simple explanation for this result: These two data streams have a small concept drift rate which does hardly change over time. Therefore, the optimal size of the case base will remain more or less constant as well. Since training data is furthermore uniformly distributed,

Table 3. Streaming classification rates

	IBL-DS	LWF02	LWF04	Win200	Win400	Win800	TWF996	TWF998
GAUSS	.807	.772	.801	.798	.771	.707	.724	.668
SINE2	.878	.827	.858	.857	.833	.758	.804	.738
DISTRIB	.939	.943	.945	.943	.940	.937	.900	.899
RANDOM	.724	.706	.721	.714	.706	.673	.654	.621
STAGGER	.909	.770	.770	.929	.914	.875	.872	.866
MIXED	.865	.691	.685	.856	.831	.758	.804	.738
HYPER2	.921	.922	.918	.913	.918	.916	.879	.875
HYPER5	.865	.855	.858	.839	.849	.844	.807	.804
MEANS2	.908	.914	.899	.885	.903	.910	.881	.881
MEANS5	.780	.800	.770	.714	.751	.783	.710	.740

using a window of fixed size is indeed a suitable strategy. Again, however, note that the classification rate of IBL-DS is not much worse.

4.3 Real World Data

So far, only synthetic data sets have been used. Even though synthetic data allows one to model special effects and, hence, is advantageous from this point of view, conducting experiments with real-world data is of course also desirable. As already mentioned above, however, real-world data streams are hard to obtain. To overcome this problem, at least to some extent, we decided to use (static) benchmark data sets from the UCI repository and to prepare them as data streams.

In this regard, note that any data set, provided it is not too small, can be considered as a stream, simply by imposing an arbitrary ordering of the data items. More specifically, however, since a data stream is by definition an open-ended sequence, an ordered data set can at best be considered as a *section* of a stream. Besides, concept drift will usually not be present in streams of that kind. To simulate concept drift we did the following, inspired by [23]: First, the data set is put into a random order. Then, the most relevant input feature is identified using the “Correlation-based Feature Subset Selection” method implemented in WEKA, and the data is sorted according to the value of that attribute. Finally, the attribute itself is deleted from the data, thereby becoming a “hidden factor”. This way, a kind of concept shift is obtained in the case of discrete attributes, whereas numerical attributes will produce gradual concept drift.

To qualify as a (pseudo-)stream, a data set should first of all not be too small. Moreover, a useful data set will have a reasonable number of classes and moreover, should not contain features that are highly correlated with the attribute used to simulate concept drift. These requirements are satisfied by only a few UCI data sets. For our studies, we selected the Balance, Car and Nursery data. Table 4 summarizes the main properties of these data sets.

Table 4. UCI data sets prepared as (pseudo-)streams

data set	#instances	#classes	#attributes	selected attribute
Balance	625	3	4	right-distance (numeric)
Car	1728	4	6	safety(nominal)
Nursery	11025	5	8	health (nominal)

IBL-DS was employed in its default parameter setting. For LWF the parameters $\beta = .04$ and $\beta = .1$ are used, TWF is run with *weight* = .99 and *weight* = .995, and the fixed sliding window approach (Win) with *size* = 50, 100, 200. To show that concept drift does really occur, the standard instance based algorithm (IBL) that simply stores all instances is additionally applied.

The results are presented in table 5. As can be seen, IBL-DS again performs very well, even for these different types of data streams, without the need to change its parameters (apart from the case base size). Moreover, the standard instance-based algorithm clearly drops off and cannot compete, probably due to the simulated drift of the concept.

Table 5. Streaming classification rates for UCI data

	Balance	Car	Nursery
IBL-DS	.805	.889	.900
LWF04	.782	.700	.899
LWF10	.846	.700	.842
Win50	.813	.819	.827
Win100	.815	.862	.856
Win200	.785	.888	.878
TWF99	.795	.889	.877
TWF995	.787	.887	.900
IBL	.693	.745	.839

5 Summary and Conclusions

We have presented an instance-based adaptive classification algorithm for learning on data streams. This algorithm, called IBL-DS, has a number of desirable properties that are not, at least not as a whole, shared by existing alternative methods. Our experiments suggest that IBL-DS is very flexible and thus able to adapt to an evolving environment quickly, a point of utmost importance in the data stream context. In particular, two specially designed editing strategies are used in combination in order to successfully deal with both gradual concept drift and abrupt concept shift. Besides, IBL-DS is relatively robust and produces good results when being used in a default setting for its parameters.

The JAVA implementation of IBL-DS is available for experimental purposes and can be downloaded, along with a documentation, under the following address: www.witi.cs.uni-magdeburg.de/iti_dke.

There are various directions for further research. For example, techniques for model (case base) maintenance and adaptation like the one proposed in [19] are quite interesting, since they are less heuristic than those currently employed in IBL-DS. Even though it is not immediately clear how such techniques can be used in a streaming application with tight time and resource constraints, investigating such approaches in more detail and trying to adapt them correspondingly seems worthwhile.

References

1. Aggarwal, C., Han, J., Wang, J., Yu, P.: A framework for clustering evolving data streams. In: Aberer, K., Koubarakis, M., Kalogeraki, V. (eds.) *Databases, Information Systems, and Peer-to-Peer Computing*. LNCS, vol. 2944, Springer, Heidelberg (2004)
2. Aha, D.W. (ed.): *Lazy Learning*. Kluwer Academic Publishers, Dordrecht (1997)
3. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. *Machine Learning* 6(1), 37–66 (1991)
4. Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and issues in data stream systems. In: *Proc. 21st ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems*, Madison, Wisconsin, pp. 1–16. ACM Press, New York (2002)
5. Ben-David, S., Gehrke, J., Kifer, D.: Detecting change in data streams. In: *Proc. VLDB-04* (2004)
6. Bercken, J., Blohsfeld, B., Dittrich, J., Krämer, J., Schäfer, T., Schneider, M., Seeger, B.: XXL - a library approach to supporting efficient implementations of advanced database queries. In: *Proceedings of the VLDB*, pp. 39–48 (2001)
7. Ciaccia, P., Patella, M., Rabitti, F., Zezula, P.: Indexing metric spaces with M-tree. In: *Proc. SEBD'97*, Verona, Italy, June 1997, pp. 67–86 (1997)
8. Cormode, G., Muthukrishnan, S.: What's hot and what's not: tracking most frequent items dynamically. In: *Proc. 22nd ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems*, pp. 296–306. ACM Press, New York (2003)
9. Dasarathy, B.V. (ed.): *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos (1991)
10. Datar, M., Muthukrishnan, S.: Estimating rarity and similarity over data stream windows. In: Möhring, R.H., Raman, R. (eds.) *ESA 2002*. LNCS, vol. 2461, pp. 323–334. Springer, Heidelberg (2002)
11. Domingos, P.: Unifying instance-based and rule-based induction. *Machine Learning* 24, 141–168 (1996)
12. Domingos, P., Hulten, G.: A general framework for mining massive data streams. *Journal of Computational and Graphical Statistics*, 12 (2003)
13. Gaber, M.M., Zaslavsky, A., Krishnaswamy, S.: Mining data streams: A review. *ACM SIGMOD Record* 34(1) (2005)
14. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Bazzan, A.L.C., Labidi, S. (eds.) *SBIA 2004*. LNCS (LNAI), vol. 3171, pp. 286–295. Springer, Heidelberg (2004)
15. Gama, J., Medas, P., Rodrigues, P.: Learning decision trees from dynamic data streams. In: Preneel, B., Tavares, S. (eds.) *SAC 2005*, pp. 573–577. ACM Press, New York (2005)

16. Golab, L., Tamer, M.: Issues in data stream management. *SIGMOD Rec.* 32(2), 5–14 (2003)
17. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 97–106. ACM Press, New York (2001)
18. Keogh, E., Kasetty, S.: On the need for time series data mining benchmarks: A survey and empirical demonstration. In: *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, July 2002, pp. 102–111. ACM Press, New York (2002)
19. Klinkenberg, R., Joachims, T.: Detecting concept drift with support vector machines. In: *Proc. ICML, 17th Int. Conf. on Machine Learning*, San Francisco, CA, pp. 487–494 (2000)
20. Klinkenberg, R.: Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis (IDA), Special Issue on Incremental Learning Systems Capable of Dealing with Concept Drift* 8(3), 281–300 (2004)
21. Kolter, J.Z., Maloof, M.A.: Dynamic weighted majority: A new ensemble method for tracking concept drift. Technical Report CSTR-20030610-3, Department of Computer Science, Georgetown University, Washington, DC (June 2003)
22. Kubat, M., Widmer, G.: Adapting to drift in continuous domains. In: Lavrač, N., Wrobel, S. (eds.) *Machine Learning: ECML-95. LNCS*, vol. 912, p. 307. Springer, Heidelberg (1995)
23. Law, Y.N., Zaniolo, C.: An adaptive nearest neighbor classification algorithm for data streams. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) *PKDD 2005. LNCS (LNAI)*, vol. 3721, Springer, Heidelberg (2005)
24. McKenna, E., Smyth, B.: Competence-guided editing methods for lazy learning. In: *ECAI*, pp. 60–64 (2000)
25. Salganicoff, M.: Tolerating concept and sampling shift in lazy learning using prediction error context switching. *Artif. Intell. Rev.* 11(1-5), 133–155 (1997)
26. Stanfil, C., Waltz, D.: Toward memory-based reasoning. *Communications of the ACM* 29, 1213–1228 (1986)
27. Tsybmal, A.: The problem of concept drift: definitions and related work. Technical Report TCD-CS-2004-15, Department of Computer Science, Trinity College Dublin, Ireland (2004)
28. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: *KDD '03. Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 226–235. ACM Press, New York (2003)
29. Widmer, G., Kubat, M.: Effective learning in dynamic environments by explicit context tracking. In: Brazdil, P.B. (ed.) *Machine Learning: ECML-93. LNCS*, vol. 667, pp. 227–243. Springer, Heidelberg (1993)
30. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. *Mach. Learn.* 23(1), 69–101 (1996)
31. Witten, I., Frank, E.: *Data Mining: Practical machine learning tools and techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Softening the Margin in Discrete SVM

Carlotta Orsenigo¹ and Carlo Vercellis²

¹ Dip. di Scienze Economiche, Aziendali e Statistiche, Università di Milano, Italy
carlotta.orsenigo@unimi.it

² Dip. di Ingegneria Gestionale, Politecnico di Milano, Italy
carlo.vercellis@polimi.it

Abstract. Discrete support vector machines are models for classification recently introduced in the context of statistical learning theory. Their distinctive feature is the formulation of mixed integer programming problems aimed at deriving optimal separating hyperplanes with minimum empirical error and maximum generalization capability. A new family of discrete SVM is proposed in this paper, for which the hyperplane establishes a variable softening of the margin to improve the separation among distinct classes. Theoretical bounds are derived to finely tune the parameters of the optimization problem. Computational tests on benchmark datasets in the biolife science application domain indicate the effectiveness of the proposed approach, that appears dominating against traditional SVM in terms of accuracy and percentage of support vectors.

Keywords: discrete support vector machines; statistical learning theory; classification; biolife sciences; data mining.

1 Introduction

Statistical learning theory (Vapnik, 1995; 1998) has germinated accurate and practical methods for classification, among which the well known family of *support vector machines* (SVM) (Burges, 1998; Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2002). The structural risk minimization (SRM) principle, which plays a central role in statistical learning theory, establishes that a good classifier trained on a given dataset should simultaneously minimize the empirical classification error and the generalization error, in order to achieve a high discrimination capability on unseen data. To attain this goal, SVM approximate the misclassification error with the sum of the slacks of the training points from the discriminant function, and the generalization error with the reciprocal of the margin of separation.

Discrete support vector machines, originally introduced in (Orsenigo and Vercellis, 2003; 2004), are a successful alternative to SVM that is based on the idea of accurately evaluating the number of misclassified examples instead of measuring their distance from the separating hyperplane. Hence, discrete SVM rely on the SRM principle, and their common distinguishing feature is the evaluation of the empirical error by a discrete function which counts the number of misclassified instances. This leads to the formulation of mixed integer programming models that are hard to solve to optimality, but for which good sub-optimal solutions can be obtained by devising

fast heuristic algorithms. Starting from the original formulation, discrete SVM have been effectively extended in several directions, to deal with multi-class problems (Orsenigo and Vercellis, 2007a) or to learn from a small number of training examples (Orsenigo and Vercellis, 2007b).

In this paper we propose a new variant of discrete SVM for which the optimal discriminating hyperplane establishes a variable softening of the margin of separation by including a new term into the objective function and modifying some of the constraints of the optimization problem. The explicit inclusion of the margin as a variable allows to regulate more effectively the trade-off between the misclassification error on the training data and the generalization capability, by means of the corresponding cost coefficient. When this latter is large, there is an advantage in taking a large margin as well, and the misclassification error increases; by converse, small values of the coefficient induce the margin to decrease, reducing the empirical error at the expense of generalization. In this respect, the proposed model has relations with the line of reasoning behind ν -SVM (Schölkopf and Smola, 2002). As for previous discrete SVM, a set of binary variables regulate the complexity of the separation rule to further improve the generalization capability of the classifier.

The description of the proposed model in section 3 highlights also a new perspective on previous discrete SVM models, by expressing the objective function as a weighted sum of 1-norms and 0-norms. To avoid numerical difficulties due to ill-conditioning, some theoretical bounds are derived which allow to finely tune the parameters of the optimization model.

Finally, in order to validate the proposed approach, computational tests on benchmark datasets taken from biolife classification problems have been systematically conducted. These problems, described in section 4, represent well-known challenging binary and multicategory classification tasks. The computational results seem to suggest that the proposed classifier dominates in terms of accuracy traditional SVM and ν -SVM with different kernels, and also improves previous versions of discrete SVM. Furthermore, the new model is characterized by a percentage of support vectors significantly smaller than for the other SVM methods considered.

The chapter is organized as follows. In section 2 we provide a definition of classification problems and a description of traditional SVM. Discrete SVM are introduced in section 3, where the new model for softening the margin is proposed. Finally, computational tests are illustrated in section 4.

2 Classification and Support Vector Machines

Classification problems require to discriminate between distinct pattern sets. Formally, m points (\mathbf{x}_i, y_i) , $i \in M = \{1, 2, \dots, m\}$, in the $(n+1)$ -dimensional real space \mathfrak{R}^{n+1} are given, where \mathbf{x}_i is a n -dimensional vector of *attributes* or *features* and y_i a scalar representing the *label* or *class* of instance i . Let $\mathcal{D} = \{1, 2, \dots, D\}$ be the set of distinct class values that can be assumed by y_i and $\mathcal{D}^* = \mathcal{D} \cup \{*\}$, where the symbol $\{*\}$ stands for an undefined predicted value. We are then required to determine a

discriminant function $f_{\alpha} : \mathfrak{X}^n \rightarrow \mathcal{D}^*$ such that a suitable measure of discrepancy between $f_{\alpha}(\mathbf{x}_i)$ and y_i is minimized over $i \in M$. Here α is a vector of adjustable parameters by which the discriminant function is indexed. It is further assumed that the m points are independently drawn from some common unknown probability distribution $P(\mathbf{x}, y)$.

To assess the accuracy of $f_{\alpha}(\mathbf{x})$, the whole set of instances is partitioned into two disjoint subsets, denoted respectively as *training* and *test* set. For a given classifier, the discriminant function is learned using only instances from the training set and then applied to predict the class of points in the test set in order to evaluate the accuracy. The attention will be restricted in what follows to binary classification problems arising when $D = 2$, i.e. the class attribute y_i takes only two different values, which may be labeled as $\{-1, +1\}$ without loss of generality.

It has been noticed that most binary classifiers actually generate as output a function $g_{\alpha} : \mathfrak{X}^n \rightarrow \mathfrak{R}$, termed *score function* or *margin*, whose sign discriminates between the two classes, so that $f_{\alpha}(\mathbf{x}) = \text{sgn}(g_{\alpha}(\mathbf{x}))$. Moreover, for these binary margin classifiers, the magnitude of the score $g_{\alpha}(\mathbf{x})$ can be viewed as a measure of confidence in the class assignment.

Let also \mathcal{A} and \mathcal{B} denote the two sets of points represented by the vectors \mathbf{x}_i in the space \mathfrak{X}^n and corresponding respectively to the two classes $y_i = -1$ and $y_i = +1$. If the two point sets \mathcal{A} and \mathcal{B} are linearly separable, that is when their convex hulls do not intersect, at least a separating hyperplane $g_{\alpha}(\mathbf{x}) = \mathbf{w}'\mathbf{x} - b$ exists which discriminates the points in \mathcal{A} from those in \mathcal{B} , i.e. such that

$$\begin{aligned} \mathbf{w}'\mathbf{x}_i - b &> 0 & \text{if } \mathbf{x}_i \in \mathcal{A} \\ \mathbf{w}'\mathbf{x}_i - b &< 0 & \text{if } \mathbf{x}_i \in \mathcal{B} \end{aligned} \quad i \in M. \quad (1)$$

In this case $f_{\alpha}(\mathbf{x}_i) = \text{sgn}(g_{\alpha}(\mathbf{x}_i)) = y_i$ for every $i \in M$, and $\alpha = (\mathbf{w}, b)$. In order to determine the coefficients $\mathbf{w} \in \mathfrak{X}^n$ and $b \in \mathfrak{R}$ a linear programming problem can be solved, as in (Mangasarian, 1965). Conversely, when the point sets \mathcal{A} and \mathcal{B} are not linearly separable, inequalities (1) cannot be satisfied for all the instances of the dataset and more complex schemes of classification should be devised to minimize a reasonable measure of violation of (1). In this perspective, a successful approach is represented by the theory of support vector machines (SVM) (Vapnik, 1995; 1998). SVM are based on the structural risk minimization (SRM) principle, that establishes the concept of reducing the empirical classification error as well as the generalization error in order to achieve a higher accuracy on unseen data. This leads to the minimization of the expression

$$\frac{\|\mathbf{w}\|_2^2}{2} + \frac{1}{2m} \sum_{i=1}^m |y_i - f_{\alpha}(\mathbf{x}_i)|, \quad \text{where } \|\mathbf{w}\|_2 = \sqrt{\sum_{j \in N} w_j^2}, \quad (2)$$

assuming conventionally that $|y_i - f_u(\mathbf{x}_i)| = 2$ whenever $f_u(\mathbf{x}_i) = *$, that is when the predicted class of \mathbf{x}_i is left undefined. The first term in (2) is the reciprocal of the *margin of separation*, defined as the distance between the pair of parallel *canonical supporting hyperplanes* $\mathbf{w}'\mathbf{x} - b - 1 = 0$ and $\mathbf{w}'\mathbf{x} - b + 1 = 0$. The geometrical interpretation of the canonical hyperplanes and the margin is given in figure 1. The maximization of this margin increases the generalization capability of the classifier. The second term in (2) is called *empirical risk* and expresses the accuracy of the classifier on the training set through the percentage of misclassified instances. The examples $\mathbf{x}_i, i \in M$, that are positioned precisely on the two canonical supporting hyperplanes are termed *support vectors*, and are in some sense more relevant in the training set than other examples, since they contribute more directly to determining the separating hyperplane. For example, in figure 1 we have three support vectors, two of them of class “white” and the third of class “black”.

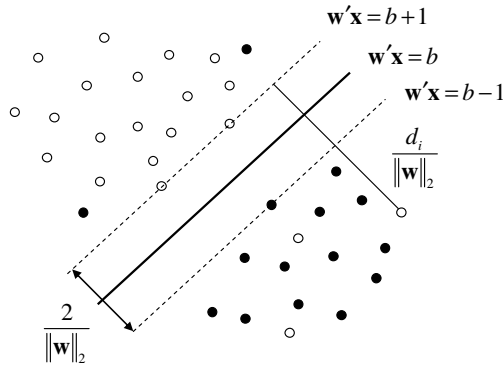


Fig. 1. Margin maximization for linearly non separable sets

Define a nonnegative continuous slack variable $d_i, i \in M$, for each instance of the dataset, such that the following linear constraints are satisfied:

$$y_i(\mathbf{w}'\mathbf{x}_i - b) \geq 1 - d_i, \quad i \in M . \tag{3}$$

Then, as noticed in (Vapnik,1995), for sufficiently small $\sigma > 0$ the function

$$F_\sigma(\mathbf{d}) = \sum_{i \in M} d_i^\sigma \tag{4}$$

evaluates the empirical risk by counting the number of misclassification errors on the training set, provided the discriminant function is defined as follows:

$$f_u(\mathbf{x}) = \begin{cases} +1 & \text{if } \mathbf{w}'\mathbf{x} - b \geq 1 \\ -1 & \text{if } \mathbf{w}'\mathbf{x} - b \leq -1 . \\ * & \text{otherwise} \end{cases} \tag{5}$$

Hence, the optimal separating hyperplane can be determined by solving the optimization problem

$$\min \frac{\lambda}{2} \|\mathbf{w}\|_2^2 + \frac{1}{m} \sum_{i \in M} d_i^\sigma \quad (6)$$

$$\text{s. to } y_i(\mathbf{w}'\mathbf{x}_i - b) \geq 1 - d_i, \quad i \in M \quad (7)$$

$$d_i \geq 0, i \in M; \mathbf{w}, b \text{ free,}$$

for sufficiently small $\sigma > 0$, where λ is a parameter available to control the trade-off between the generalization capability of the classifier and the misclassification error. To avoid computational difficulties, in the classical theory of SVM the empirical risk is regularized and approximated by solving problem (6) only for the value $\sigma = 1$. This leads to a quadratic programming problem, whose solution is obtained via Lagrangean duality which, beside the computational benefits, also provides the interpretation of the support vectors. Taking advantage of the dual formulation and of suitable kernel functions (Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2002), SVM proceed by projecting the original examples into a higher dimensional feature space, in which the linear separation is derived, allowing to efficiently obtain nonlinear discriminations in the original space.

3 Softening the Margin in Discrete Support Vector Machines

A different family of classification models, termed *discrete support vector machines*, has been introduced in (Orsenigo and Vercellis, 2003; 2004) and is motivated by a more strict adherence to the SRM principle, by counting the number of misclassified examples instead of measuring their distance from the separating hyperplane. The distinctive trait of discrete SVM is the accurate representation of the empirical error, by using the total misclassification error in the objective function, in place of the sum of the slacks considered in (6) by traditional SVM. Hence, the rationale behind discrete SVM is that a precise evaluation of the empirical error could possibly lead to a more accurate classifier.

In this section we extend previous discrete SVM models by proposing a different mathematical programming formulation for determining the optimal separating hyperplane. To derive the objective function we modify the expression given in (2). First, the 2-norm of \mathbf{w} is replaced by the 1-norm

$$\|\mathbf{w}\|_1 = \sum_{j \in N} |w_j|. \quad (8)$$

Beside computational advantages, this choice also removes asymmetries between the two terms in (2), as the margin is squared whereas the misclassification error is not. As the second term we utilize the total misclassification rate. Finally, a third term is added to (2) in order to minimize the number of nonzero components of the coefficients vector \mathbf{w} , that is the number of features utilized for discrimination. This choice has two main motivations: by reducing the number of used features, it is likely

to increase the generalization capability of a model learned on the training set. Furthermore, the induced rules become simpler and more suitable to the interpretation of domain experts; see also (Orsenigo and Vercellis, 2006). The third term we introduce in (2) is given by the following zero-norm

$$\|\mathbf{w}\|_0^0 = \text{card}(i : w_i \neq 0), \quad (9)$$

where $\text{card}(E)$ is the cardinality of a set E . The zero-norm can be seen as the limit of the p -th power of the ℓ_p -norm

$$\|\mathbf{w}\|_p^p = \left(\sum_{j \in N} w_j^p \right)^{1/p}, \quad (10)$$

as in (Weston et al., 2003). The resulting objective function is a weighted combination of three components, with weights $(\beta_1, \beta_2, \beta_3)$ regulating their trade-off:

$$\frac{\beta_1}{2} \|\mathbf{w}\|_1 + \frac{\beta_2}{2m} \sum_{i=1}^m |y_i - f_a(\mathbf{x}_i)| + \beta_3 \|\mathbf{w}\|_0^0. \quad (11)$$

If we let $\theta_i = \theta(\mathbf{x}_i)$ be an indicator function defined as

$$\theta_i = \begin{cases} 0 & \text{if } f_a(\mathbf{x}_i) = y_i \\ 1 & \text{if } f_a(\mathbf{x}_i) \neq y_i \end{cases} \quad i \in M, \quad (12)$$

then (11) can be expressed as a weighted sum of norms

$$\frac{\beta_1}{2} \|\mathbf{w}\|_1 + \frac{\beta_2}{m} \|\boldsymbol{\theta}\|_0^0 + \beta_3 \|\mathbf{w}\|_0^0. \quad (13)$$

By assuming the discriminant function given in (5), the following discrete support vector machine optimization problem can be formulated to determine the optimal separating hyperplane

$$\min_{\mathbf{w}, b, L \subseteq M} \frac{\beta_1}{2} \|\mathbf{w}\|_1 + \frac{\beta_2}{m} \|\boldsymbol{\theta}\|_0^0 + \beta_3 \|\mathbf{w}\|_0^0 \quad (14)$$

$$\text{s. to} \quad y_i(\mathbf{w}'\mathbf{x}_i - b) \geq 1, \quad i \in L.$$

In problem (14) one is required to simultaneously determine a subset $L \subseteq M$ of points to be left misclassified and the hyperplane coefficients \mathbf{w}, b in a way to minimize the weighted sum of the reciprocal of the margin, the empirical error and the number of active features. Model (14) is a complex combinatorial optimization problem defined over an exponentially high number of subsets. However, it can be transformed into a mixed integer programming model by properly defining binary variables. Indeed, the number of misclassified points is already counted by the binary indicator variables $\theta_i, i \in M$, whereas the following binary variables account for the nonzero components of the vector \mathbf{w}

$$\tau_j = \begin{cases} 0 & \text{if } w_j = 0 \\ 1 & \text{if } w_j \neq 0 \end{cases} \quad j \in N. \quad (15)$$

Let $c_i, i \in M$, denote the misclassification cost associated to instance i , and $p_j, j \in N$, the penalty cost for using attribute j . Let also Q and P be sufficiently large constant values. By linearizing the 1-norm in (13) with nonnegative bounding variables $u_j, j \in N$, problem (14) can thus be formulated as the following mixed binary linear programming problem (DSVM):

$$\min \frac{\beta_1}{2} \sum_{j \in N} u_j + \frac{\beta_2}{m} \sum_{i \in M} c_i \theta_i + \beta_3 \sum_{j \in N} p_j \tau_j \quad (\text{DSVM})$$

$$\text{s. to} \quad y_i(\mathbf{w}'\mathbf{x}_i - b) \geq 1 - Q\theta_i, \quad i \in M \quad (16)$$

$$-u_j \leq w_j \leq u_j, \quad j \in N \quad (17)$$

$$u_j \leq P\tau_j, \quad j \in N \quad (18)$$

$$\theta_i \in \{0, 1\}, i \in M; u_j \geq 0, \tau_j \in \{0, 1\}, j \in N; \mathbf{w}, b \text{ free}.$$

Model (DSVM) is a mixed binary linear optimization problem, notoriously more difficult to solve to optimality than continuous linear optimization. However, it can be solved by means of an efficient heuristic procedure, based on a sequence of linear optimization problems, for obtaining suboptimal solutions. Model (DSVM) can be used as a linear perceptron; alternatively, it can be framed within a recursive procedure for the generation of oblique classification trees, to derive an optimal separating hyperplane at each node of the tree, as in (Orsenigo and Vercellis, 2003; 2004). In the quoted references, it was shown by means of extensive testing that the increase in model complexity is justified by a more accurate discrimination and a higher generalization capability, due to the correct estimation of the empirical misclassification error and the minimization of the number of attributes defining the separating hyperplane.

Here we wish to go one step further in the evaluation of the misclassification error. Observe first that the formulation of the discriminant function $f_u(\mathbf{x})$ implies that all points falling between the pair of canonical supporting hyperplanes are considered unclassified. Correspondingly, constraints (7) in the SVM formulation stick on the same assumption, since they determine a strictly positive slack variable d_i , and therefore determine a misclassification error. The same remains true also in the (DSVM) formulation, as constraints (16) imply $\theta_i = 1$ whenever point \mathbf{x}_i lies between the two canonical hyperplanes. In light of this assumption, these models are termed *soft margin* classifiers, because the region between the canonical hyperplanes is left unclassified. The width of the margin between the two canonical hyperplanes is driven by the first term in (DSVM), expressing the norm of \mathbf{w} . In this paper we want

to exploit the effect of a different classification model, for which the margin is determined by means of the explicit inclusion of a new variable ε . To do this we consider the following discriminant function

$$f_a(\mathbf{x}) = \begin{cases} +1 & \text{if } \mathbf{w}'\mathbf{x} - b \geq \varepsilon \\ -1 & \text{if } \mathbf{w}'\mathbf{x} - b \leq -\varepsilon \end{cases}, \quad (19)$$

where $\varepsilon > 0$ is a variable to be determined. Consequently, the empirical error appearing as the second term in the bound (2) should account only for those points which are actually misclassified by the discriminant separating function. We therefore formulate the ε -discrete support vector machine (ε -DSVM) model:

$$\min \quad \frac{1}{2} \sum_{j \in N} u_j + \frac{1}{m} \sum_{i \in M} c_i \theta_i + \mu \sum_{j \in N} p_j \tau_j - \nu \varepsilon \quad (\varepsilon\text{-DSVM})$$

$$\text{s. to} \quad y_i(\mathbf{w}'\mathbf{x}_i - b) \geq \varepsilon - Q\theta_i, \quad i \in M \quad (20)$$

$$-u_j \leq w_j \leq u_j, \quad j \in N \quad (21)$$

$$u_j \leq P\tau_j, \quad j \in N \quad (22)$$

$$\theta_i \in \{0, 1\}, i \in M; u_j \geq 0, \tau_j \in \{0, 1\}, j \in N; \varepsilon \geq \rho; \mathbf{w}, b \text{ free},$$

where $\rho > 0$ is a lower threshold to prevent the case $\varepsilon = 0$. The variable ε can be interpreted as a way of progressively softening or hardening the separation between the two classes determined by the optimal hyperplane: when ε decreases and approaches 0 the soft margin around the separating hyperplane reduces and tend to vanish, whereas the opposite is true when ε increases.

Notice that the objective function of model (ε -DSVM) still incorporates the first term related to margin maximization and the third term for minimizing the number of active features, so that the generalization capability on new data should be preserved. However, the explicit inclusion of the variable ε allows to fix to the value $\beta_1 = \beta_2 = 1$ of the weight parameters for the first two terms. The trade-off between the misclassification error on the training data and the generalization capability is indeed regulated in model (ε -DSVM) by the parameter ν : when ν is large, there is an advantage in taking ε large as well, and the misclassification error increases. By converse, small values of ν induce ε to decrease, reducing the empirical error at the expense of generalization. In this respect, model (ε -DSVM) has relations with the line of reasoning behind ν -SVM (Schölkopf and Smola, 2002). As for (DSVM), the parameter μ in (ε -DSVM) weighs the complexity of the separation rule.

One might be tempted to solve problem (ε -DSVM) also for $\varepsilon = 0$. Notice however that in this case constraints (20) might lead to optimal solutions for which $\mathbf{w} = \mathbf{0}$. Efforts have been devoted by some authors to prevent the risk of incurring in such degenerate separating hyperplanes; a survey is given in (Koehler and Erenguc,

1990). Formulation (DSVM) does not disallow trivial solutions in principle; however degenerate solutions may occur solely if $\mathbf{w} = \mathbf{0}$ actually represents the optimal solution to the classification concept, due to the presence of the binary variables θ_i with their misclassification costs c_i . Indeed, a degenerate hyperplane with $\mathbf{w} = \mathbf{0}$ corresponds to labeling all instances with the same class, and arises therefore when any separating hyperplane (with $\mathbf{w} \neq \mathbf{0}$) causes the objective function in (DSVM) to increase. Constraints (20), combined with the inclusion into the objective function of the variable ε , prevent trivial solutions for problem (ε -DSVM).

Models (DSVM) and (ε -DSVM) include two big constants Q and P required to force binary variables to 1. When such feature occurs in a mixed integer programming model, it is well known that the constants should be chosen as smallest as possible in order to prevent numerical difficulties during the solution phase, due to ill-conditioning in the coefficients matrix. Hence, in the sequel we will provide tight bounds on the values that parameters Q and P should assume in order to ensure proper forcing of the binary variables. We start by limiting parameter Q . Let R denote the smallest radius of a sphere enclosing all the points of the training dataset. We have

$$2R = \max_{\mathbf{x}_i, \mathbf{x}_j} \sqrt{\sum_{k=1}^m (x_{ik} - x_{jk})^2}. \quad (23)$$

Let also

$$s_i = \mathbf{w}'\mathbf{x}_i - b, \quad i \in M, \quad (24)$$

be the slack of point \mathbf{x}_i with respect to the separating hyperplane. Without loss of generality we can assume that the slack of each point $\mathbf{x}_i, i \in M$, with respect to an optimal hyperplane in model (ε -DSVM) is bounded above in modulus by the maximum distance between pairs of points in the training set, that is $|s_i| \leq 2R$. This maximum corresponds to classifying all points in the training set as belonging to the same class. Furthermore, due to constraints (20), parameter Q must be large enough that

$$\varepsilon - y_i(\mathbf{w}'\mathbf{x}_i - b) \leq Q, \quad i \in M. \quad (25)$$

We then want to show that the choice $Q = 2R + 1$ is sufficient to guarantee that conditions (25) are met. Indeed

$$\varepsilon - y_i(\mathbf{w}'\mathbf{x}_i - b) \leq 1 + |y_i(\mathbf{w}'\mathbf{x}_i - b)| \leq 1 + |\mathbf{w}'\mathbf{x}_i - b| = 1 + |s_i| \leq 1 + 2R. \quad (26)$$

The limitation of parameter P follows a different line of reasoning. It is known (Vapnik, 1995) that imposing the condition $\|\mathbf{w}\|_2 \leq A$ implies that the separating hyperplane cannot be closer than $1/A$ to any of the training points \mathbf{x}_i . We therefore impose that $P = A/\sqrt{n}$, where n is the number of features of the classification problem, and A is selected as a satisfactory level of closeness between the separating hyperplane and the training points. In light of constraints (17) this choice leads to the required limitation:

$$\|\mathbf{w}\|_2 = \sqrt{\sum_{j \in N} w_j^2} \leq \sqrt{\sum_{j \in N} u_j^2} \leq A. \quad (27)$$

Here is a short description of the heuristic procedure for determining a feasible suboptimal solution to model (ε -DSVM), based on a sequence of linear programming (LP) problems. The heuristic starts by considering the LP relaxation of problem (ε -DSVM). Each LP problem (ε -DSVM)_{*t*+1} in the sequence is obtained by fixing to zero the relaxed binary variable with the smallest fractional value in the optimal solution of the predecessor (ε -DSVM)_{*t*}. Notice that, if problem (ε -DSVM)_{*t*} is feasible and its optimal solution is integer feasible, the procedure is stopped, and the solution generated at iteration *t* is retained as an approximation to the optimal solution of problem (ε -DSVM). Otherwise, if problem (ε -DSVM)_{*t*+1} is unfeasible, the procedure modifies the previous LP problem (ε -DSVM)_{*t*} by fixing to 1 all of its fractional variables. Problem (ε -DSVM)_{*t*+1} defined in this way is feasible and any of its optimal solutions is integer. Thus, the procedure is stopped and the solution found for (ε -DSVM)_{*t*} is retained as an approximation to the optimal solution of (ε -DSVM).

4 Computational Tests

To validate the effectiveness of model (ε -DSVM) some computational tests were performed on three benchmark datasets, each referring to a different biolife application domain. Two of these datasets, “promoter gene sequences” (*Promoter*) and “splice junction gene sequences” (*Splice*), are available from the UCI Machine Learning Repository (Hettich et al., 1998). The third dataset, indicated as *Structure* in the sequel, is obtained by collecting the samples proposed in (Ding and Dubchak, 2001). In particular, the *Promoter* dataset consists of 106 examples each represented by 57 sequential nucleotide positions, which may take one of the values in the set {a, c, g, t}. Each example is labeled with the class {+} if it represents a promoter, that is a gene sequence with a biological promoter activity, and with the class {-} if it is given by a non-promoter sequence. The problem is to discriminate between promoters, which initiate the process of gene expression, and non-promoter gene sequences. The *Splice* dataset contains 3175 examples each represented by 60 sequential nucleotide positions which, as for the *Promoter* dataset, take their values in the set {a, c, g, t}. In this case, each sequence may belong to one of three different classes, according to the inclusion of a splicing site. More specifically, the generic example may be an “acceptor site” (IE), a “donor site” (EI) or neither of them, and the problem is to discriminate between acceptors and donors in the presence of imperfect domain theory. Finally, the *Structure* dataset is composed by 698 protein sequences which derive by the union of two samples. The first one, collected by (Dubchak et al., 1999) and generally used for the training process, consists of 313 proteins having no more than 35% identity with each other. The second dataset, utilized as an independent test sample, is the PDB-40D set developed by the authors of the SCOP database (Andreeva et al., 2004; Murzin et al., 1995). It contains 385 proteins possessing less than 40% of the sequence identity and having less than 35% identity with the proteins

contained in the first dataset. The proteins in these two samples are associated to their secondary structural class, which takes one of the following values: α (α -helix secondary structure), β (β -sheet secondary structure), α/β (mixed or alternating α -helix and β -sheet segments) and $\alpha+\beta$ (α -helix and β -sheet segments not mixed). Here the problem is to predict the secondary structure of a protein. The number of classes, attributes and examples for each dataset is given in table 1.

Table 1. Description of the datasets

Dataset	Description				
	classes	examples	attributes	(training,tuning)	prediction
Promoter	2	106	57	(60, 14)	32
	(+, -)		numerical attributes	50% class + 50% class -	50% class + 50% class -
Splice	3	3175	60	(600, 100)	1000
	(IE, EI, N)		numerical attributes	24% class IE 24% class EI 52% class N	24% class IE 24% class EI 52% class N
Structure	4	698	83	(214, 99)	385
	(α , β , α/β , $\alpha+\beta$)		numerical attributes	18% class α 34% class β 37% class α/β 11% class $\alpha+\beta$	16% class α 30% class β 37% class α/β 17% class $\alpha+\beta$

Eight alternative classification approaches were selected for comparison to model (ε -DSVM): discrete SVM (DSVM), (ε -LSVM), which is obtained by replacing in the (ε -DSVM) model the misclassification rate with the sum of the slacks of the misclassified examples, and six methods derived by combining SVM and ν -SVM with linear (SVM_{LIN}), Gaussian (SVM_{GAUSS}) and radial basis function (SVM_{RBF}) as kernels. The results for classifiers (ε -DSVM), (DSVM) and (ε -LSVM) were obtained using the sequential LP-based heuristic described in section 3, whereas the computations for SVM methods were achieved by means of LIBSVM library (Chang and Lin, 2001). In order to perform the multicategory classification of *Splice* and *Structure*, all the methods were framed within the *round robin* scheme (Allwein et al., 2000; Orsenigo and Vercellis, 2007a). Moreover, in applying all the classifiers the attributes describing each dataset were converted into numerical explanatory variables. In particular, for the *Promoter* dataset the generic nucleotide position $s_i \in \{a,c,g,t\}$ was replaced by the conditional probability of observing the symbol s_i given the positive promoter status of a gene sequence. Notice that the same could be done for the non-promoter class value. For the *Splice* dataset, which leads to a multicategory classification task, each nucleotide position $s_i \in \{a,c,g,t\}$ was replaced by the corresponding numeric value in the set $\{1,2,3,4\}$. Finally, for the *Structure* dataset four sets of numerical attributes were used for representing the amino acids sequences: these are amino acids composition (20 attributes), predicted secondary

structure (21 attributes), hydrophobicity (21 attributes) and polarity (21 attributes). This choice was motivated by the fact that on the same dataset these attributes exhibited a notable explanatory power (Ding and Dubchak, 2001; Orsenigo and Vercellis, 2007c).

For evaluating the performance of the competing methods we proceeded in the following way. Each dataset was divided into three subsets, representing respectively the training, the tuning and the prediction set. For *Promoter* and *Splice* this last sample was extracted from the original dataset, whereas for *Structure* we used the out-of-sample dataset described above. Table 1 indicates for each dataset the size of the samples which share the same composition of the corresponding original datasets in terms of class values representatives. Then we applied *holdout estimation* (Kohavi, 1995) using the training and the tuning sets, in order to properly regulate the parameters of each classification models and assess their accuracy on the past examples. The accuracy values as well as the average computational time required for the training process are indicated in tables 2 and 3.

Table 2. Comparison among ε -DSVM, SVM and discrete SVM: holdout accuracy (%), prediction accuracy (%) on the out-of-sample datasets, computational time for training (sec)

Dataset	Method				
	ε -DSVM	DSVM	SVM _{LIN}	SVM _{RBF}	SVM _{GAUSS}
Promoter	100	92.9	92.9	92.9	92.9
	100	96.9	100	100	100
	0.5	0.5	0.3	0.3	0.3
Splice	86.0	83.0	65.0	74.0	66.0
	83.9	83.1	74.7	77.7	76.3
	12	7	1	1	1
Structure	88.9	86.9	85.9	83.8	88.9
	76.9	75.3	75.1	74.5	75.8
	3	3	0.3	0.3	0.3

Table 3. Comparison among ε -DSVM, ε -LSVM and ν -SVM: holdout accuracy (%), prediction accuracy (%) on the out-of-sample datasets, computational time for training (sec)

Dataset	Method				
	ε -DSVM	ε -LSVM	ν -SVM _{LIN}	ν -SVM _{RBF}	ν -SVM _{GAUSS}
Promoter	100	100	100	100	100
	100	100	100	100	100
	0.5	0.2	0.3	0.3	0.3
Splice	86.0	75.0	67.0	70.0	64.0
	83.9	80.7	75.6	79.2	77.5
	12	0.5	1	0.8	0.5
Structure	88.9	85.9	85.9	85.9	83.8
	76.9	75.1	75.1	75.1	76.6
	3	1	0.3	0.3	0.3

Finally, we applied the optimal classification function obtained by holdout estimation on the prediction datasets, to the end of investigating the discriminatory ability of each classifier on future unseen examples. The prediction accuracy achieved on the out-of-sample datasets is shown in tables 2 and 3.

From the results presented in tables 2 and 3 we can draw the empirical conclusion that model (ε -DSVM) represents an effective classification method, since it is able to achieve the highest accuracy for all the datasets considered in our tests. This remark holds true for the holdout estimation as well as for the out-of-sample datasets classification. Notice that the higher accuracy achieved in the training process, which might lead to overfitting, allows model (ε -DSVM) to reach a higher precision in predicting the class value of future examples, therefore achieving also a high generalization capability. Moreover, it is worth to observe that the optimal value of the variable ε controlling the softening of the margin in model (ε -DSVM) is far from 1, since it ranges in the interval $[0.1, 0.4]$ for the three datasets considered in our tests.

To further explore the usefulness of model (ε -DSVM) we counted the number of support vectors (SVs) corresponding to the optimal separating hyperplane of each competing method. Tables 4 and 5, which contain the percentage of examples that each classifier used as support vectors for performing the discrimination, show that the optimal separating function obtained by means of (ε -DSVM) is consistently based on a smaller number of SVs. This means that the classification rules generated by the proposed method are more robust and capable of a higher generalization capability.

Table 4. Support vectors (%) for ε -DSVM, SVM and discrete SVM

Dataset	Method				
	ε -DSVM	DSVM	SVM _{LIN}	SVM _{RBF}	SVM _{GAUSS}
Promoter	8.3	26.7	100	100	100
Splice	7.8	13.3	36.0	56.2	40.5
Structure	44.4	44.9	44.9	73.8	51.9

Table 5. Support vectors (%) for ε -DSVM, ε -LSVM and ν -SVM

Dataset	Method				
	ε -DSVM	ε -LSVM	ν -SVM _{LIN}	ν -SVM _{RBF}	ν -SVM _{GAUSS}
Promoter	8.3	43.3	35.0	35.0	31.6
Splice	7.8	18	37.3	69.0	46.0
Structure	44.4	62	44.9	59.3	47.7

References

- Allwein, E., Schapire, R., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research* 1, 113–141 (2000)
- Andreeva, A., Howorth, D., Brenner, S.E., Hubbard, T.J., Chothia, C., Murzin, A.G.: SCOP database in 2004: refinements integrate structure and sequence family data. *Nucleic Acids Res.* 32, D226–D229 (2004)
- Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* 2, 121–167 (1998)
- Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines (2001)
- Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000)
- Ding, C.H., Dubchak, I.: Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics* 17, 349–358 (2001)
- Dubchak, I., Muchnik, I., Mayor, C., Dralyuk, I., Kim, S.H.: Recognition of a protein fold in the context of the Structural Classification of Proteins (SCOP) classification. *Proteins* 35, 401–407 (1999)
- Hettich, S., Blake, C., Merz, C.: UCI repository of machine learning databases. (1998), <http://www.ics.uci.edu/mlearn/MLRepository.html>
- Koehler, G.J., Erenguc, S.: Minimizing misclassifications in linear discriminant analysis. *Decision Sciences* 21, 63–85 (1990)
- Kohavi, R.: A study of cross-validation and bootstrapping for accuracy estimation and model selection. In: *Proc. of the 14th International Joint Conference on Artificial Intelligence*, pp. 338–345. Morgan Kaufmann, San Francisco (1995)
- Mangasarian, O.L.: Linear and nonlinear separation of patterns by linear programming. *Operations Research* 13, 444–452 (1965)
- Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C.: SCOP: a structural classification of protein database for the investigation of sequence and structures. *J. Mol. Biol.* 247, 536–540 (1995)
- Orsenigo, C., Vercellis, C.: Multivariate classification trees based on minimum features discrete support vector machines. *IMA Journal of Management Mathematics* 14, 221–234 (2003)
- Orsenigo, C., Vercellis, C.: Discrete support vector decision trees via tabu-search. *Journal of Computational Statistics and Data Analysis* 47, 311–322 (2004)
- Orsenigo, C., Vercellis, C.: Rule induction through discrete support vector decision trees. In: Triantaphyllou, E., Felici, G. (eds.) *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques*, pp. 305–325. Springer, Heidelberg (2006)
- Orsenigo, C., Vercellis, C.: Multicategory classification via discrete support vector machines. *Computational Management Science* (in press) (2007a)
- Orsenigo, C., Vercellis, C.: Accurately learning from few examples with a polyhedral classifier. *Computational Optimization and Applications* (in press) (2007b)
- Orsenigo, C., Vercellis, C.: Protein folding classification through multicategory discrete SVM. In: Felici, G., Vercellis, C. (eds.) *Mathematical Methods for Knowledge Discovery and Data Mining*, Idea Group, USA (in press) (2007c)
- Schölkopf, B., Smola, A.J.: *Learning with kernels. Support vector machines, regularization, optimization and beyond*. MIT Press, Cambridge (2002)
- Vapnik, V.: *The nature of statistical learning theory*. Springer, Heidelberg (1995)
- Vapnik, V.: *Statistical Learning Theory*. Wiley, Chichester (1998)
- Weston, J., Elisseeff, A., Schölkopf, B., Tipping, M.: Use of the Zero-Norm with Linear Models and Kernel Methods. *Journal of Machine Learning Research* 3, 1439–1461 (2003)

Feature Selection Using Ant Colony Optimization (ACO): A New Method and Comparative Study in the Application of Face Recognition System

Hamidreza Rashidy Kanan, Karim Faez, and Sayyed Mostafa Taheri

Image Processing and Pattern Recognition Lab, Electrical Engineering Department,
Amirkabir University of Technology (Tehran Polytechnic), Hafez Avenue, Tehran, Iran, 15914
{rashidykanan, kfaez, mostafa_taheri}@aut.ac.ir

Abstract. Feature Selection (FS) and reduction of pattern dimensionality is a most important step in pattern recognition systems. One approach in the feature selection area is employing population-based optimization algorithms such as Genetic Algorithm (GA)-based method and Ant Colony Optimization (ACO)-based method. This paper presents a novel feature selection method that is based on Ant Colony Optimization (ACO). ACO algorithm is inspired of ant's social behavior in their search for the shortest paths to food sources. Most common techniques for ACO-Based feature selection use the priori information of features. However, in the proposed algorithm, classifier performance and the length of selected feature vector are adopted as heuristic information for ACO. So, we can select the optimal feature subset without the priori information of features. This approach is easily implemented and because of using one simple classifier in it, its computational complexity is very low. Simulation results on face recognition system and ORL database show the superiority of the proposed algorithm.

Keywords: Feature Selection, Ant Colony Optimization (ACO), Genetic Algorithm, Face Recognition.

1 Introduction

Several parameters can affect the performance of pattern recognition system. Among them, feature extraction and representation of patterns can be considered as a most important. Reduction of pattern dimensionality via feature extraction and selection belongs to the most fundamental step in data processing [1].

Feature Selection (FS) is extensive and spread across many fields, including document classification, data mining, object recognition, biometrics, remote sensing and computer vision [2]. Given a feature set of size n , the FS problem is to find a minimal feature subset of size m ($m < n$) while retaining a suitably high accuracy in representing the original features. In real word problems FS is a must due to the abundance of noisy, irrelevant or misleading features [3].

As a simplest way, the best subset of features can be found by evaluating all the possible subsets, which is known as exhaustive search. This procedure is quite

impractical even for a moderate size feature set. Because the number of feature subset combinations with m features from a collection of n ($m < n$, $m \neq 0$) feature is $n!/[m!(n-m)!]$ and the total number of these combinations is $(2^n - 2)$.

For most practical problems, an optimal solution can only be guaranteed if a monotonic criterion for evaluating features can be found, but this assumption rarely holds in the real-world [4]. As a result, we must find solutions which would be computationally feasible and represent a trade-off between solution quality and time.

Usually FS algorithms involve heuristic or random search strategies in an attempt to avoid this prohibitive complexity. However, the degree of optimality of the final feature subset is often reduced [3].

Among too many methods which are proposed for FS, population-based optimization algorithms such as Genetic Algorithm (GA)-based method and Ant Colony Optimization (ACO)-based method have attracted a lot of attention. These methods attempt to achieve better solutions by using knowledge from previous iterations.

Genetic algorithms (GA's) are optimization techniques based on the mechanics of natural selection. They used operations found in natural genetics to guide itself through the paths in the search space [5]. Because of their advantages, recently, GA's have been widely used as a tool for feature selection in pattern recognition.

Metaheuristic optimization algorithm based on ant's behavior (ACO) was represented in the early 1990s by M. Dorigo and colleagues [6]. ACO is a branch of newly developed form of artificial intelligence called Swarm Intelligence. Swarm intelligence is a field which studies "the emergent collective intelligence of groups of simple agents" [7]. In groups of insects which live in colonies, such as ants and bees, an individual can only do simple task on its own, while the colony's cooperative work is the main reason determining the intelligent behavior it shows [8].

ACO algorithm is inspired of ant's social behavior. Ants have no sight and are capable of finding the shortest route between a food source and their nest by chemical materials called pheromone that they leave when moving.

ACO algorithm was firstly used in solving Traveling Salesman Problem (TSP) [9]. Then has been successfully applied to a large number of difficult problems like the Quadratic Assignment Problem (QAP) [10], routing in telecommunication networks, graph coloring problems, scheduling and etc. This method is particularly attractive for feature selection as there seems to be no heuristic that can guide search to the optimal minimal subset every time [3]. In the other hand, if features are represented as a graph, ant will discover best feature combinations as they traverse the graph.

In this paper a new modified ACO-Based feature selection algorithm has been introduced. The classifier performance and the length of selected feature vector are adopted as heuristic information for ACO. So proposed algorithm needs no priori knowledge of features. Proposed algorithm is applied to two different feature subsets that are Pseudo Zernike Moment Invariant (PZMI) and Discrete Wavelet Transform (DWT) Coefficients in the application of face recognition system and finally the classifier performance and the length of selected feature vector are considered for performance evaluation.

The rest of this paper is organized as follows. Section 2 presents a brief overview of feature selection methods. Ant Colony Optimization (ACO) and Genetic Algorithm (GA) are described in Sections 3 and 4 respectively. Section 5 explains the proposed feature

selection algorithm and finally, Sections 6 and 7 attain the experimental results and conclusion.

2 An Overview of Feature Selection (FS) Approaches

Feature selection algorithms can be classified into two categories based on their evaluation procedure [11]. If an algorithm performs FS independently of any learning algorithm (i.e. it is a completely separate preprocessor), then it is a filter approach (open-loop approach). This approach is based mostly on selecting features using between-class separability criterion [11]. If the evaluation procedure is tied to the task (e.g. classification) of the learning algorithm, the FS algorithm employs the wrapper approach (closed-loop approach). This method searches through the feature subset space using the estimated accuracy from an induction algorithm as a measure of subset suitability.

The two mentioned approaches are also classified into five main methods which they are Forward Selection, Backward elimination Forward/Backward Combination, Random Choice and Instance based method.

FS methods may start with no features, all features, a selected feature set or some random feature subset. Those methods that start with an initial subset usually select these features heuristically beforehand. Features are added (Forward Selection) or removed (Backward Elimination) iteratively and in the Forward/Backward Combination method features are either iteratively added or removed or produced randomly thereafter.

The disadvantage of Forward Selection and Backward Elimination methods is that the features that were once selected/eliminated cannot be later discarded/re-selected. To overcome this problem, Pudil et al. [12] proposed a method to flexibly add and remove features. This method has been called floating search method.

In the wrapper approach the evaluation function calculates the suitability of a feature subset produced by the generation procedure and compares this with the previous best candidate, replacing it if found to be better. A Stopping criterion is tested every iteration to determine whether the FS process should continue or not.

Other famous FS approaches are based on the Genetic Algorithm (GA) [13], Simulated Annealing [3] and Ant Colony Optimization (ACO) [3, 8, 14, 15, 16].

[14] has proposed a hybrid approach for speech classification problem. This method has used combination of mutual information and ACO. [15] has used the hybrid of ACO and mutual information for selection of features in the forecaster. [16] has utilized the Fisher Discrimination Rate (FDR) as a heuristic information in the ACO-Based feature selection method which is used for selection of network intrusion features. [3] has used a ACO for finding rough set reducts. [8] has introduced a Ant-Miner which is used a difficult pheromone updating strategy and state transition rule.

Also, some surveys of feature selection algorithms are given in [1, 17, 18].

3 Ant Colony Optimization (ACO)

In the early 1990s, ant colony optimization (ACO) was introduced by M. Dorigo and colleagues as a novel nature-inspired metaheuristic for the solution of hard combinatorial optimization (CO) problems [19]. ACO belongs to the class of metaheuristics, which are approximate algorithms used to obtain good enough solutions to hard CO problems in a reasonable amount of computation time [19].

The ability of real ants to find shortest routes is mainly due to their depositing of pheromone as they travel; each ant probabilistically prefers to follow a direction rich in this chemical. The pheromone decays over time, resulting in much less pheromone on less popular paths. Given that over time the shortest route will have the higher rate of ant traversal, this path will be reinforced and the others diminished until all ants follow the same, shortest path (the "system" has converged to a single solution) [3].

In general, an ACO algorithm can be applied to any combinatorial problem as far as it is possible to define:

- ❖ Appropriate problem representation. The problem must be described as a graph with a set of nodes and edges between nodes.
- ❖ Heuristic desirability (η) of edges. A suitable heuristic measure of the "goodness" of paths from one node to every other connected node in the graph.
- ❖ Construction of feasible solutions. A mechanism must be in place whereby possible solutions are efficiently created.
- ❖ Pheromone updating rule. A suitable method of updating the pheromone levels on edges is required with a corresponding evaporation rule. Typical methods involve selecting the n best ants and updating the paths they chose.
- ❖ Probabilistic transition rule. The rule that determines the probability of an ant traversing from one node in the graph to the next.

3.1 ACO for Feature Selection

The feature selection task may be reformulated into an ACO-suitable problem. ACO requires a problem to be represented as a graph. Here nodes represent features, with the edges between them denoting the choice of the next feature. The search for the optimal feature subset is then an ant traversal through the graph where a minimum number of nodes are visited that satisfies the traversal stopping criterion. Figure 1 illustrates this setup. The ant is currently at node *a* and has a choice of which feature to add next to its path (dotted lines). It chooses feature *b* next based on the transition rule, then *c* and then *d*. Upon arrival at *d*, the current subset $\{a; b; c; d\}$ is determined to satisfy the traversal stopping criterion (e.g. a suitably high classification accuracy has been achieved with this subset). The ant terminates its traversal and outputs this feature subset as a candidate for data reduction.

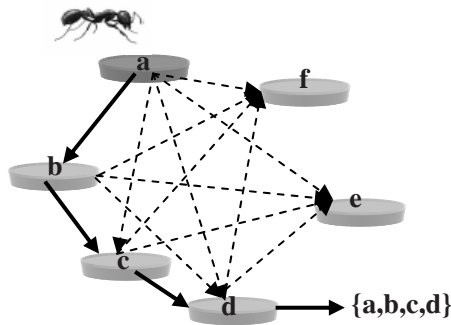


Fig. 1. ACO problem representation for FS

A suitable heuristic desirability of traversing between features could be any subset evaluation function - for example, an entropy-based measure [19], rough set dependency measure [20] or the Fisher Discrimination Rate (FDR)[16]. The heuristic desirability of traversal and edge pheromone levels are combined to form the so-called probabilistic transition rule, denoting the probability of an ant at feature i choosing to travel to feature j at time t :

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} & \text{if } j \in J_i^k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Where k is the number of ants, η_{ij} is the heuristic desirability of choosing feature j when at feature i (η_{ij} is optional but often needed for achieving a high algorithm performance [21]), J_i^k is the set of neighbor nodes of node i which have not yet been visited by the ant k . $\alpha > 0$, $\beta > 0$ are two parameters that determine the relative importance of the pheromone value and heuristic information (the choice of α , β is determined experimentally) and $\tau_{ij}(t)$ is the amount of virtual pheromone on edge (i,j) .

The overall process of ACO feature selection can be seen in figure 2. The process begins by generating a number of ants, k , which are then placed randomly on the graph (i.e. each ant starts with one random feature). Alternatively, the number of ants to place on the graph may be set equal to the number of features within the data; each ant starts path construction at a different feature. From these initial positions, they traverse edges probabilistically until a traversal stopping criterion is satisfied. The resulting subsets are gathered and then evaluated. If an optimal subset has been found or the algorithm has executed a certain number of times, then the process halts and outputs the best feature subset encountered. If neither condition holds, then the pheromone is updated, a new set of ants are created and the process iterates once more.

The pheromone on each edge is updated according to the following formula:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta \tau_{ij}(t) \quad (2)$$

Where:

$$\Delta \tau_{ij}(t) = \sum_{k=1}^n (\mathcal{Y}'(S^k) / |S^k|) \quad (3)$$

This is the case if the edge (i,j) has been traversed; $\Delta \tau_{ij}(t)$ is 0 otherwise. The value $0 \leq \rho \leq 1$ is decay constant used to simulate the evaporation of the pheromone, S^k is the feature subset found by ant k . The pheromone is updated

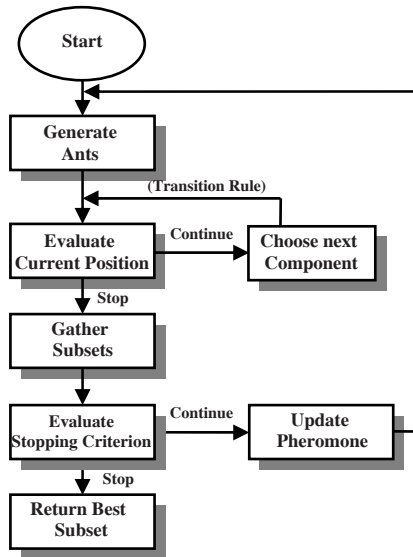


Fig. 2. ACO-based feature selection overview

according to both the measure of the "goodness" of the ant's feature subset γ' and the size of the subset itself. By this definition, all ants update the pheromone.

4 Genetic Algorithm (GA)

The GA's is a stochastic global search method that mimics the metaphor of natural biological evolution [5]. These algorithms are general purpose optimization algorithms with a probabilistic component that provide a means to search poorly understood, irregular spaces.

GA's work with a population of points rather than a single point. Each "point" is a vector in hyperspace representing one potential (or candidate) solution to the optimization problem. A population is, thus, just an ensemble or set of hyperspace vectors. Each vector is called a chromosome in the population. The number of elements in each vector (chromosome) depends on the number of parameters in the optimization problem and the way to represent the problem. How to represent the problem as a string of elements is one of the critical factors in successfully applying a GA (or other evolutionary algorithm) to a problem.

A typical series of operations carried out when implementing a GA paradigm is:

- ❖ Initialize the population;
- ❖ Calculate fitness for each chromosome in population;
- ❖ Reproduce selected chromosomes to form a new population;
- ❖ Perform crossover and mutation on the population;
- ❖ Loop to second step until some condition is met.

Initialization of the population is commonly done by seeding the population with random values. The fitness value is proportional to the performance measurement of the function being optimized. The calculation of fitness values is conceptually simple. It can, however, be quite complex to implement in a way that optimizes the efficiency of the GA's search of the problem space. It is this fitness that guides the search of the problem space.

After fitness calculation, the next step is reproduction. Reproduction comprises forming a new population, usually with the same total number of chromosomes, by selecting from members of the current population using a stochastic process that is weighted by each of their fitness values. The higher the fitness, the more likely it is that the chromosome will be selected for the new generation. One commonly used way is a "roulette wheel" procedure that assigns a portion of a roulette wheel to each population member where the size of the portion is proportional to the fitness value. This procedure is often combined with the elitist strategy, which ensures that the chromosome with the highest fitness is always copied into the next generation.

The next operation is called crossover. To many evolutionary computation practitioners, crossover is what distinguishes a GA from other evolutionary computation paradigms. Crossover is the process of exchanging portions of the strings of two "parent" chromosomes. An overall probability is assigned to the crossover process, which is the probability that given two parents, the crossover process will occur. This probability is often in the range of 0.65–0.80. The final operation in the typical GA procedure is mutation. Mutation consists of changing an element's value at random, often with a constant probability for each element in the population. The probability of mutation can vary widely according to the application and the preference of the person exercising the GA. However, values of between 0.001 and 0.01 are not unusual for mutation probability.

4.1 GA for Feature Selection

Several approaches exist for using GAs for feature subset selection. The two main methods that have been widely used in the past are as follow. First is due to [13], of finding an optimal binary vector in which each bit corresponds to a feature (Binary Vector Optimization method (BVO)). A '1' or '0' suggests that the feature is selected or dropped, respectively. The aim is to find the binary vector with the smallest number of 1's such that the classifier performance is maximized. This criterion is often modified to reduce the dimensionality of the feature vector at the same time [21]. The second and more refined technique [22]) uses an m-ary vector to assign weights to features instead of abruptly dropping or including them as in the binary case. This gives a better search resolution in the multidimensional space [23].

5 Proposed Feature Selection Algorithm

The main steps of proposed algorithm are as follows:

1) Initialization

- Determine the population of ants (p).
- Set the intensity of pheromone trail associated with any feature.
- Determine the maximum of allowed iterations (k)

2) Generation ants and evaluation of each ants

- Any ant (A_i , $i=1:p$) randomly is assigned to one feature and it should visit all features and build solutions completely. In this step, the evaluation criterion is Mean Square Error (MSE) of the classifier. If any ant could not decrease the MSE of the classifier in three successive steps, it finished its work and exit.

3) Evaluation of the selected subset of each ant

- In this step the importance of the selected subset of each ant is evaluated through classifier performance. Then the subsets according to their MSE are sorted and some of them are selected according to ACS and AS_{rank} algorithms.

4) Check the stop criterion

- If the number of iterations is more than the maximum allowed iteration exit, otherwise continue.

5) Pheromone updating

- For features which are selected in the step 3 pheromone intensity are updated.

6) Go to 2 and continue

6 Experimental Results

To show the utility of proposed feature selection algorithm and to compare with GA-Based approach two sets of experiments were carried out. For experimental studies we have considered ORL gray scale face image database. This database contains 400 facial images from 40 individuals in different states. So, the number of classes in our experiments is 40. The total number of images in each class is 10.

Figure 3 shows some samples images of this database.



Fig. 3. Some samples of ORL database

Two different sets of features were extracted from each face image which they are Pseudo Zernike Moment Invariant (PZMI) and Discrete Wavelet Transform (DWT)

Coefficient. Then proposed ACO-based and GA-based feature selection methods are applied to each feature set and finally, the length of selected feature vector and classifier performance are considered for evaluating the proposed algorithm.

The details of experiments are as follows:

6.1 Feature Extraction

After preprocessing (histogram equalization) of facial images, we extract the PZMI and DWT coefficients as a feature vector.

In the PZMI feature extraction step, the PZMI of orders 1 to 20 and their repetitions are extracted from any face image. I.e. for any n (order of PZMI), we extract one feature vector of order n and all repetitions m ($m \leq n$). For example if we choose $n = n_0$, we have one feature vector which has $(n_0 + 1)$ elements [24, 25].

In the DWT feature extraction step, Discrete Wavelet Transform is applied to any face images. Since the face images are not continuous, we used Haar wavelet which is also discrete. We applied pyramid algorithm to each preprocessed image for decomposing it into 3 resolution levels. Then we used the approximation of images at level 3 and converted them into vectors by concatenating the columns. Dimensions of ORL database images is 92×112 , so after decomposing them, the length of wavelet feature for each image is 168 [26].

For scale invariancy of extracted features (PZMI and DWT Coefficients), we normalized them.

6.2 Feature Selection

After the extraction of PZMI and DWT Coefficients, ACO and GA are used to select the optimal feature sets.

We consider our system as a block diagram that is shown in Figure 4.

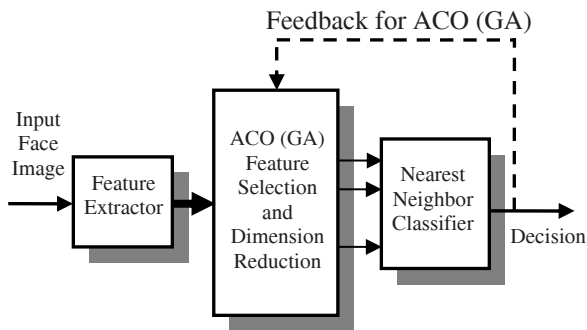


Fig. 4. Block Diagram of proposed feature selection scheme

For GA-based feature selector, we set the length of chromosomes to L which $L=20$ for PZMI features and $L=168$ for DWT Coefficients features. Each gene

$g_i (i = 1, 2, \dots, L)$ corresponds to a specific order of PZMI or specific DWT Coefficient feature component.

If $g_i = 1$, this means we select this order (this feature component) as one of optimal orders (optimal components). Otherwise, $g_i = 0$ means discard it. Because most of orders (feature components) may be selected, the probability of every bit being equal to 1 is set to 0.8 when the initial population of chromosomes is creating. Its purpose is to speed up the convergence.

Given a chromosome q the fitness function $F(q)$ is defined as:

$$F(q) = \frac{1}{\sum_{x \in \Omega} \delta(x, q)} \quad (4)$$

Here Ω is the training image set for GAs and $\delta(x, q)$ is defined as:

$$\delta(x, q) = \begin{cases} 1, & \text{if } x \text{ is classified correctly} \\ 0, & \text{if } x \text{ is misclassified} \end{cases} \quad (5)$$

For simplicity, we have used the nearest neighbor classifier and the aim is to find a binary vector with the smallest number of 1's such that the classifier performance is maximized. In order to select the individuals for the next generation, GA's roulette wheel selection method was used.

Further Genetic Algorithms parameters are summarized in Table 1.

Table 1. GA-Based Feature Selection Parameters

	PZMI Features	DWT Coefficients Features
Population size	50	50
Number Of Generation	25	25
Chromosome length	20	168
Probability of crossover	0.7	0.7
Probability of mutation	0.003	0.003

For ACO-Based feature selector, we use same primary features which are utilized in the GA-Based feature selector.

In this step, we have applied proposed algorithm to the extracted features in the formats of ACS and AS_{rank} with the same parameters.

Various parameters for leading to better convergence are tested and the best parameters that are obtained by simulations are as follows:

$\alpha=1$, $\beta=0.1$, $\rho=0.2$, the initial pheromone intensity of each feature is equal to 1, the number of Ant in every iteration $p=50$ and the maximum number of iterations $k=25$. These values are chosen to justify the comparison with GA. Selected features of each method are classified using nearest neighbor classifier and the obtained MSE is considered for performance evaluation.

6.2.1 Comparison of ACO-Based and GA-Based Methods

The results of this step are summarized in Tables 2 and 3.

Table 2 gives, for each method, the best MSE and the average of execution time.

Table 2. MSE and execution time of Three Different Methods

Method	MSE		Time (s)	
	PZMI	DWT	PZMI	DWT
GA	3.5%	2%	1080	1560
ACO (ACS)	3%	1%	780	1320
ACO (AS_{rank})	1.5%	0.25%	300	960

Both ACO-Based methods (ACS and AS_{rank}) produce much lower classification errors and execution times than GA-Based method. ACS and AS_{rank} algorithms have comparable performance. The ACS method is faster than AS_{rank} method however it has lower performance. Also, Table 3 gives the optimal selected features for each method. Both ACO-Based and GA-Based methods significantly reduce the number of original features. But ACO-Based method (ACS and AS_{rank}) chooses fewer features.

Table 3. Selected Features of Three Different Methods

Method	Selected Features		Number of Selected Features	
	PZMI (Order)	DWT (Component)	PZMI	DWT
GA	2, 4, 8, 10, 12, 13	1, 4, 5, 6, 12, 14, 15, 17, 21, 26, 29, 32, 35, 37, 40, 44, 47, 49, 52, 53, 57, 58, 62, 64, 69, 72, 74, 79, 84, 88, 91, 93, 98, 100, 107, 111, 117, 125, 136, 137, 139, 145, 149, 155, 161, 167, 168	55	47
ACO (ACS)	4, 6, 9, 12, 13	4, 5, 20, 25, 29, 30, 37, 42, 44, 49, 58, 62, 68, 70, 73, 93, 94, 95, 96, 100, 102, 109, 112, 113, 114, 118, 120, 121, 125, 132, 138, 139, 141, 147, 149, 152, 156, 157, 158, 159, 163, 168	49	42
ACO (AS_{rank})	6, 8, 10, 14	2, 6, 18, 21, 22, 42, 49, 57, 58, 73, 75, 83, 93, 95, 96, 100, 116, 118, 122, 125, 136, 138, 144, 147, 149, 153, 157, 158, 160, 167	42	30

Tables 2 and 3 show that using proposed method, we can achieve 99.75% and 98.5% recognition rate only with 30 and 42 selected features for DWT coefficients and PZMI features respectively.

Since feature selection is typically done in an off-line manner, the execution time of a specific algorithm is of much less importance than its ultimate classification performance. So, we can say that the AS_{rank} method gives better results. Finally, selected features of each method are classified and the obtained recognition rates are shown in Figures 5 and 6.

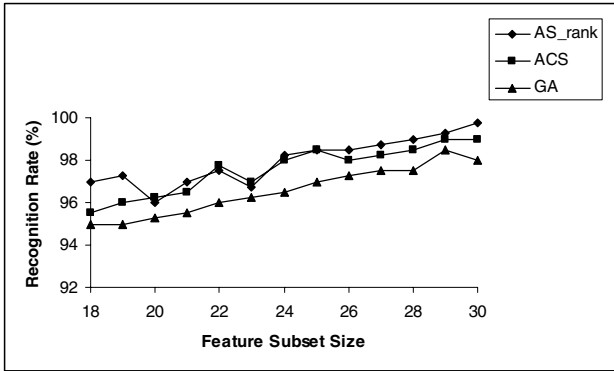


Fig. 5. Recognition Rate of DWT Feature Subsets Obtained Using AS_{rank} , ACS and GA

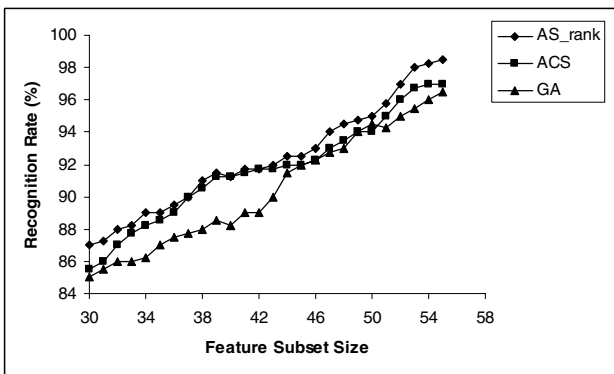


Fig. 6. Recognition Rate of PZMI Feature Subsets Obtained Using AS_{rank} , ACS and GA

It can be seen that the performance of both ACS and AS_{rank} was found to be much better than that of GA-Based method and proposed ACO-Based algorithm was able to achieve better performance than GA-Based algorithm in most of the cases.

7 Conclusion

In this paper a novel ACO-Based feature selection algorithm is presented. In the proposed algorithm, the classifier performance and the length of selected feature vector are adopted as heuristic information for ACO. So, we can select the optimal feature subset without the priori knowledge of features. Proposed approach is simulated in the ACS and AS_{rank} algorithm formats. To show the utility of proposed algorithm and to compare with GA-Based approach two sets of experiments were carried out on two different sets of features that they are PZMI and DWT coefficients. Simulation results on face recognition system and ORL database show that the proposed ACO-Based method outperforms GA-Based method since, it achieved better performance with the lower number of features and execution time.

Acknowledgment

This research was supported by the Iran Telecommunication Research Center (ITRC).

References

1. kml, L., Kittler, J.: Feature set search algorithms. In: Chen, C.H. (ed.) Pattern Recognition and Signal Processing, Sijhoff and Noordhoff, the Netherlands (1978)
2. Ani, A.A.: An Ant Colony Optimization Based Approach for Feature Selection. In: Proceeding of AIML Conference (2005)
3. Jensen, R.: Combining rough and fuzzy sets for feature selection. Ph.D. Thesis, University of Edinburgh (2005)
4. Kohavi, R.: Feature Subset Selection as search with Probabilistic Estimates. AAAI Fall Symposium On Relevance (1994)
5. Srinivas, M., Patnik, L.M.: Genetic Algorithms: A Survey. IEEE Computer Society Press, Los Alamitos (1994)
6. Dorigo, M., Caro, G.D.: Ant Colony Optimization: A New Meta-heuristic. In: Proceeding of the Congress on Evolutionary Computing (1999)
7. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York (1999)
8. Liu, B., Abbass, H.A., McKay, B.: Classification Rule Discovery with Ant Colony Optimization. IEEE Computational Intelligence 3(1) (2004)
9. Dorigo, M., Maniezzo, V., Colorni, A.: The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B 26(1), 29–41 (1996)
10. Maniezzo, V., Colorni, A.: The Ant System Applied to the Quadratic Assignment Problem. Knowledge and Data Engineering 11(5), 769–778 (1999)
11. Duda, R.O., Hart, P.E.: Pattern Recognition and Scene Analysis. Wiley, Chichester (1973)
12. Pudil, P., Novovicova, J., Kittler, J.: Floating search methods in feature selection. Pattern Recognition Letters 15, 1119–1125 (1994)
13. Siedlecki, W., Sklansky, J.: A note on genetic algorithms for large-scale feature selection. Pattern Recognition Letters 10(5), 335–347 (1989)

14. Ani, A.A.: Ant Colony Optimization for Feature Subset Selection. *Transactions On Engineering, Computing And Technology* 4 (2005)
15. Zhang, C.K., Hu, H.: Feature Selection Using The Hybrid Of Ant Colony Optimization and Mutual Information For The Forecaster. In: *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics* (2005)
16. Gao, H.H., Yang, H.H., And Wang, X.Y.: Ant Colony Optimization Based Network Intrusion Feature Selection And Detection. In: *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics* (2005)
17. Bins, J.: Feature Selection of Huge Feature Sets in the Context of Computer Vision. Ph.D. Dissertation, Computer Science Department, Colorado State University (2000)
18. Siedlecki, W., Sklansky, J.: On Automatic Feature Selection. *International Journal of Pattern Recognition and Artificial Intelligence* 2(2), 197–220 (1988)
19. Dorigo, M., Blum, C.: Ant colony optimization theory: A survey. *Theoretical Computer Science* 344, 243–278 (2005)
20. Pawlak, Z.: *Rough Sets: Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishing, Dordrecht (1991)
21. Yang, J., Honavar, V.: Feature Subset Selection Using a Genetic Algorithm. *IEEE Intelligent Systems* 13, 44–49 (1998)
22. Punch, W.F., Goodman, E.D., Pei, L.C.S.M., Hovland, P., Enbody, R.: Further research on Feature Selection and Classification using Genetic Algorithms. In: *Proc. Int. Conf. Genetic Algorithms*, pp. 557–564 (1993)
23. Raymer, M., Punch, W., Goodman, E., Kuhn, L., Jain, A.K.: Dimensionality Reduction Using Genetic Algorithms. *IEEE Transactions on Evolutionary Computing* 4, 164–171 (2000)
24. Rashidy Kanan, H., Faez, K., Ezoji, M.: Face Recognition: An Optimized Localization Approach and Selected PZMI Feature Vector Using SVM Classifier. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) *ICIC 2006*. LNCS, vol. 4113, pp. 690–696. Springer, Heidelberg (2006)
25. Rashidy Kanan, H., Faez, K., Ezoji, M.: An Efficient Face Recognition System Using a New Optimized Localization Method. In: *ICPR'2006*. Proceeding of the 18th International Conference on Pattern Recognition (2006)
26. Rashidy Kanan, H., Faez, K.: ZMI and Wavelet Transform Features and SVM Classifier in the Optimized Face Recognition system. In: *ISSPIT 2005*. Proceeding of the 5th IEEE International Symposium on Signal Processing and Information Technology, pp. 295–300. IEEE Computer Society Press, Los Alamitos (2005)

Outlier Detection with Streaming Dyadic Decomposition

Chetan Gupta¹ and Robert Grossman²

¹Dept. Of Mathematics, Statistics and Computer Science, University Of Illinois,
Chicago and Hewlett Packard Labs

²Open Data Partners

Abstract. In this work we introduce a new algorithm for detecting outliers on streaming data in \mathbf{R}^n . The basic idea is to compute a dyadic decomposition into cubes in \mathbf{R}^n of the streaming data. Dyadic decomposition can be obtained by recursively bisecting the cube the data lies in. Dyadic decomposition obtained under streaming setting is understood as streaming dyadic decomposition. If we view the streaming dyadic decomposition as a tree with a fixed maximum (and sufficient) size (depth), then outliers are naturally defined by cubes that contain a small number of points in the cube itself or the cube itself and its neighboring cubes. We discuss some properties of detecting outliers with streaming dyadic decomposition and we present experimental results over real and artificial data sets.

1 Introduction

Detecting outliers is an important data mining task. In this paper, we are concerned with outlier detection using a streaming data model in which we examine each point once and must detect outliers independent of the stream length.

As streaming data has become more ubiquitous, the problem of detecting outliers on streams has become more important. To give two examples: For example in case of network data, we need to be able to predict the attack patterns in real time over streaming data. In our experimental section we present results over such a data set. Other examples include highway data. If the data is being collected through sensors we need to be able to detect the traffic disruptions in real time as the data streams in.

There are several different definitions of outliers, and quite a few different approaches to detecting them. Some of these are described in the next section.

Roughly speaking, a dyadic decomposition divides \mathbf{R}^n into cubes. A cube may be further divided into sub-cubes, by splitting the edge in each dimension in half. This decomposition produces a collection of dyadic cubes at different scales. It also produces a tree, which we call a dyadic tree. Each node u of the dyadic tree is associated with a dyadic cube C_u , and a node u is child of another node v in case the corresponding cube C_u arises by dividing each edge of the cube C_v in half, as described above. A formal definition is given in Section 3.

One natural definition of an outlier is to say a point p in a data set D is an outlier if it is in the cube C_u of a leaf node u . In this case we say that the point p is an *outlier defined by a dyadic decomposition* or ODD. More generally, we can define k -outliers if the cube contains k points or less.

This generalization is useful in identifying members of a minority class.

As a simple example consider a set of points in a 2-D plane. Enclose all the points in a square, divide the square into four smaller squares, such that you could think of them as four quadrants. If all but one point lie in in Quadrants 1, 2, 3 and only one point lies in Quadrant 4, it is considered an outlier. To present a more trivial example with ordinal variables, consider a set of all tennis players till 1986, with two variables, Wimbledon winners or not and age less than 18 or not. There are lot of players who are less than 18, a lot of players who have won the Wimbledon, but only Boris Becker won it under 18. If the data is plotted in the above manner in a square with four smaller squares, there will be a box with only Boris Becker in it and he is an outlier.

Wherever a large amount of data is streaming in and there is a need to detect anomalous patterns, our approach can prove useful. The other advantage of our approach is that it can handle both the scenarios in a streaming setting: individual outliers or identifying members of a minority class. Our approach is simple, intuitive and works well with both continuous and categorical variables. We have experimented with various data sets and some of the results are presented in the paper. This approach can be extended to identify cluster centers in a streaming setting but due to lack of space we do not present those results here.

The organization of this work is as follows: In Section 2 we give an overview of the existing related work. In Section 3 we define dyadic decompositions. Section 4 we present the algorithm for computing ODDs. Section 5 contains some discussion and Section 6 is experimental studies. Section 6 contains the conclusion.

2 Related Work

Outlier detection is a problem considered in statistics and data mining. In Knorr's [1] work, a point p is defined as an outlier with respect to parameter k and λ , if no more than k points are a distance λ or less from p . In Ramaswamy's [2] work, an outlier is defined as: Given a k and a n , a point p is an outlier if the distance to its $k - th$ nearest neighbor is smaller than the corresponding value for no more than $n - 1$ other points. Breunig [3] gives every point a LOF (Local Outlier Factor), which measures how isolated an object is from its surrounding. They assign a degree to every point of being an outlier. They give an example to illustrate the importance of looking at the local rather than the global neighborhood. Since our approach is multiscale in nature, the "global" and "local" change with scale and our approach like [3] takes care of local and global outliers. The above techniques are not presented for streaming data and ours is a streaming data algorithm.

A dynamic programming approach to finding deviants (outliers in a sense) is presented by Jagdish [4]. Muthu [5] uses this idea to construct a near

optimal algorithm in a streaming setting for univariate streams. They extend this as a heuristic for a multivariate setting. Another method in streaming clustering/outlier detection is TECNO-STREAMS [6].

Our approach could be understood as a multi-resolution approach. Wavelets are popular in multi-resolution approaches. WaveCluster [7] is a grid-based clustering method that uses wavelet transform to filter the data. Scale-based clustering has been presented before. One idea is scale space clustering. Chakravarty, in [8] uses Radial Basis Function Network(RBFN) for scale-based clustering. In Wong's [9] work a statistical mechanic-based approach is used where the temperature is the scaling parameter. Another interesting work along similar lines is that of Leung [10] which attempts to provide a unified framework for various scale space approaches. Roberts [11] uses a scale-based smoothing function to estimate probability density function. Our algorithm shows similar properties to these scale-based algorithms. Our way of scaling is different since it is based on dyadic decomposition. Another work is using fractal dimensions is that of Barbara [12].

We can also look at this as a grid based approach. An early grid-based clustering algorithm is that of Warnekar [13]. They address the issue of "neighbors" in a grid setting. Two more grid based techniques are Bang [14] and GRIDCLUST [15] which also create hierarchical clustering using grids. GRIDCLUST is also in a scale-based grid clustering algorithm. Their grid structure is based on a k-d tree. In GRIDCLUST first the cells are arranged in order of their density and merged in that order. Clusters at different scale can be merged also. It is a bottom up approach. Our grid structure is different and we use a top down approach where the idea is to study each scale and see if certain parts of data should be studied at that or a finer scale. DBSCAN [16] is also a similar approach, but it is primarily a clustering algorithm and is not a multiscale dyadic cube approach. We have used data sets from Cure [17] and Chameleon [18]. Again none of these approaches have been presented in context of streaming data.

Recently, clustering algorithms for large data sets and streaming data sets have been developed. BIRCH (Balanced Iterative Reducing and Clustering) [19] clusters large data sets by using specialized tree structures to work with out-of-memory data. CLARANS (Clustering Large Applications based on RANDOM Search) [20] identifies candidate cluster centroids through analysis of repeated random samples of the original data. The k-mediod problem is a variation of k-means. Guha, et al. [21] have presented streaming algorithms using local clustering to solve the k-mediod problem. Additional work was done by Bradley, et al. [22] and some of the improvements were made by Farnstrom [23]. Aggarwal [24] presents a method for clustering high dimension data using projections. Outlier detection could be treated as a byproduct of these clustering approaches.

3 Dyadic Decompositions

In this section we provide formal definitions of dyadic decompositions and outliers defined by dyadic decompositions (ODDs). We begin by collecting and restating some standard definitions involving cubes and dyadic decompositions [25].

Definition 1. Dyadic decompositions are defined recursively as follows. Let the data set be $D \subseteq \mathbf{R}^n$. Let $|D| = N$ denote the number of points in D . First, we enclose the n -dimensional data set D in a cube. The scale of this initial cube is defined to be one. We define additional cubes in our decomposition recursively as follows. For an integer $k > 1$, we divide a cube whose scale is k , into 2 or more cubes whose scale we define to be $k + 1$, by bisecting one or more edges of the larger cube. This produces 2^c new cubes, where the number of bisections c satisfies: $1 \leq c \leq n$. We continue this process until a stopping criterion is reached, such as a cube contains fewer than a specified number of points, say, ϵ , which may be scale dependent in the sense that $\epsilon = \epsilon(k)$. We define this to be dyadic decomposition.

Definition 2. The cubes obtained during dyadic decomposition are called dyadic cubes.

Definition 3. The dyadic decomposition produces a dyadic tree, where each node u is associated with cube C_u , and a node u in the tree is child of another node v when the corresponding cube C_u arises by dividing one or more edges of the cube C_v in half.

We close this section with a remark:

Remark 1. Note that a cube in 2-dimensions (a square) has 8 possible adjacent cubes, and a cube in 3-dimensions has 26 possible neighboring cubes. In general, there are $3^d - 1$ adjacent cubes for a cube in d -dimensions.

4 Computing the Streaming Dyadic Tree Associated with a Data Set

In a streaming setting since we can look at each point only once and since we have fixed space the tree that we built can only be of a fixed depth. Building a dyadic decomposition as described above requires us to know the whole data set, i.e., before we can divide the cube containing the data set we need to know the dimensions of the cube the data set lies in. This means that we cannot have a top down construction as discussed above. This motivates the construction of streaming dyadic trees.

Let the data stream D be a sequence of points p_1, p_2, \dots in \mathbf{R}^n . Fix the maximum depth of a tree, an integer, $r_{max} \geq 0$. Let u_0 denote the root of a dyadic tree T and with a slight abuse of notation, let C_{u_0} denote the corresponding dyadic cube. If u is a node in T , and C_u is the corresponding dyadic cube, let $Count_u$ denote the number of points in the cube C_u .

The idea behind constructing a streaming dyadic tree is simple. Roughly speaking, take the first two points in the stream and build a cube that contains both points. If a new point comes in and it is outside the current cube keep doubling the cube (doubling will result in new cubes) while maintaining the depth of the tree till the new point is in a region covered by the new cubes.

This is Step 4. If a new point lies in a region already covered by cubes the point travels to the smallest cube that could contain the point. This is Step 3, Case 1. If the leaf is maximum depth increment the count of the leaf and discard the point, this is Step 3, Case 2, otherwise split the node making a new leaf for the point. This is Step 3, Case 3. The precise formulation is presented below.

For given a data stream, the decomposition obtained by the following algorithm is called a *streaming dyadic decomposition*. Associate a tree T with this decomposition.

Algorithm 1. Computing a Streaming Dyadic Tree

1. Let C_0 denote a cube that contains the point p_1 . Set $C = C_0$.
2. For $i \geq 2$, score the point p_i using T as follows. Let C_{u_0} denote the cube associated with the root node u_0 . Set $C = C_{u_0}$ and proceed to the next step.
3. If p_i is in the cube $C = C_u$ associated with the node u of T , then check each of the following three cases in order:
 - Case 1. In this case, we are at an interior node and will continue processing the point p_i . More precisely, (1) The node u has children, which divide the cube C_u into sub-cubes C_{u_1}, C_{u_2}, \dots , with corresponding nodes u_1, u_2, \dots and (2) the nodes u_i are not at the maximum depth r_{\max} . In this case, p_i is in one of the sub-cubes, say C_{u_j} . Set $C = C_{u_j}$, increment the count of the cube C_{u_j} and goto Step 3.
 - Case 2. In this case, we have reached the maximum depth of the tree and we will discard the point p_i . More precisely, (1) The node u has children, which divide the cube C_u into sub-cubes C_{u_1}, C_{u_2}, \dots , with corresponding nodes u_1, u_2, \dots and (2) the nodes u_i are at the maximum depth r_{\max} . In this case, p_i is in one of the sub-cubes, say C_{u_j} . Increment the counter associated with the node u_j and discard the point p_i . Goto Step 2.
 - Case 3. In this case, we add a new leaf to the tree containing the point p_i . More precisely, if the cube C has no children (i.e., is a leaf) and the node u is not at the maximum depth r_{\max} , then split the node u to produce sub-cubes C_{u_1}, C_{u_2}, \dots . Note that in this case, by the construction, the node C_u contains a least one point prior to the arrival of the point p_i . Also note, as mentioned, that the new leaf contains the point p_i . Goto Step 2.
4. Else, if p_i is not in the C_{u_0} , double the cube C_{u_0} until the point p_i is contained in it. Each time, the root is doubled, maintain the maximum depth of T by merging all the leaves with their parents as required. Process the point p_i , by letting $C = C_{u_0}$ and going to Step 3.

The tree obtained in the above algorithm is called a *streaming dyadic tree* associated with a data set D .

4.1 Outliers Associated with Streaming Dyadic Decompositions

Using the dyadic tree associated with a stream, we can now define outliers.

Let D be a data stream and T_{sD} the associate streaming dyadic tree.

Definition 4. A outlier associated with a streaming dyadic decomposition or ODD is a point that i) is contained in cube C_u associated with a leaf node u of the tree T_{sD} ; and ii) the cube C_u contains precisely one point. We say the ODD is of depth d in case the leaf C_u is at depth d from the root.

The above definition can be extended to a k -outlier associated with a dyadic decomposition or k -ODD by stipulating that the cube C_u contain at-most k points.

This concept can prove useful in certain practical problems where two or more points are close to each other and away from rest of the points.

Our approach has two key ideas:

1. In our approach we do a streaming division of space into dyadic cubes. If a point lies outside the currently bounded space we can keep doubling the space (while maintaining the depth of the tree) along all the dimensions till the point is inside the bounds. If it lies inside the bounds we either find a path to a leaf or we can keep subdividing the cubes till either the point lies in its own cube or the maximum depth is reached and further subdivision is not permissible.
2. Since a streaming setting means a restriction of space we save space by storing just those points that are possible ODDs. That is why only leaves of the streaming dyadic tree, T_{sD} , can store points. All points which are stored in the leaves are candidate ODDs. At the end of the stream we just need to look at the leaves containing point(s) to obtain our list of ODDs.

We close this section with few remarks:

Remark 2. We can relate this definition of an outlier to Knorr's definition of an outlier. A point p is defined as an *Knorr outlier* with respect to parameters k and λ , if no more than k points are a distance λ or less from p . We see that ODDs at depth d are similar in spirit to Knorr outliers but have a natural scale associated with them and use (empty or sparse) dyadic decompositions at scale d instead of a distance λ .

Remark 3. We can put restrictions on the space in which a point can exist inside a dyadic cube. This can result in more restrictive definition of an ODD.

Remark 4. The depth of the tree needs to be fixed in advance. It is easy to see that more the depth more nodes are available in the tree, increasing the number of candidate ODDs.

4.2 Complexity

The algorithm to building a streaming dyadic tree is designed to work for streaming data. Hence the space and time requirements are severe. Let the tree contain a total N_t nodes and leaves at any time t . N_t is bounded by fixing the maximum depth of the tree. The worst case scenario for any operation is traversing the complete tree for a worst case complexity of $O(N_t)$.

The space required is to store these nodes, $O(N_t)$ and the candidate ODDs.

Pruning is an effective solution for saving time and space. The tree can be pruned without losing any information. Once all the children of a particular node contain more than one element that node can be considered full. If a point falls within the bounds of that node it cannot be an outlier and it is discarded. This obviously saves the computation involved in further travelling down that branch of the tree. This does not affect the accuracy of the results.

In our experiments, even with a data sets of two million points in eight dimensions space was never an issue since typically, the data clusters naturally to a few branches of the tree.

5 Experimental Results

We have completed some experimental studies on artificial and real data sets using the algorithm described above.

The only variable the user needs to input is the maximum depth of the tree. We want to see how many outliers we are able to identify as ODDs.

To test the algorithm we did four series of experiments.

1. 2-D data sets: The first series of experiments used synthetic 2-D data sets.
2. The second series of experiments used large synthetic 2-D data sets.
3. The third series used data set containing computer network alerts.
4. The fourth series used data sets of NASA shuttle data.

For some experiments we have computed two measures, *Sensitivity* and *Specificity* and their sum *Goodness*.

$$Sensitivity = \frac{True\ Positives}{Total\ Positives} \quad (1)$$

$$Specificity = 1 - \frac{False\ Positives}{Data\ Set\ Size} \quad (2)$$

$$Goodness = Sensitivity + Specificity \quad (3)$$

Note that declaring all the points as outliers gives *sensitivity* as one but *specificity* is almost zero making the total *goodness* almost one. Not declaring any point as an outlier makes *specificity* one but the *sensitivity* is zero, making the *goodness* again one. Randomly declaring any point as an outlier with probability half also leads to a *goodness* of one. Therefore, any improvement over one is an improvement in terms of outlier detection. If an algorithm is indiscriminate in labelling points as outliers, and in the process wrongly labels a lot of points as outliers, the *sensitivity* might be high but the *specificity* goes down. On the other hand if an algorithm declares too few points as outliers, false positives would decrease and *specificity* will be high but the *sensitivity* will go down. In our experiments, as the maximum value for the depth of the tree r_{max} is increased the *sensitivity* goes up but the *specificity* goes down.

The algorithm is designed for a streaming setting. To simulate a streaming setting we read one record at a time from a file stored on a drive and discarded the point after that. Some of the real data sets that we have used contain categorical variables. We converted them to a set of binary variables using the simple technique of converting each category as yes/no value and adding it as an attribute.

5.1 2-D Data Sets

These are the Cure [17] and Chameleon [18] data sets which have non-spherical clusters and some outlier points distributed through the plane. In Figure 5.1 (Please see the last page) for the three individual figures, we have colored points identified as ODDs with the color blue. It is easy to see from the figures that in general what would appear as an outlier to the naked eye is also identified as an ODD by our algorithm. The first three data sets have 8000 points and the last one has 10000 points.

5.2 Large Data Sets

We created 30 large synthetic data sets. They were of three types, consisting of ten sets each:

1. The first type was in four dimensions and had 100,000 points. The data was distributed in four clusters and 1, . . . , 10 percent of the points (meaning 1000 to 10,000) were distributed randomly to create ten data sets respectively. The random points are expected to be the outliers our algorithm identifies as ODDs.
2. The second type was in four dimensions and had 1,000,000 points. The data was distributed as with the previous experiment.
3. The third type was in eight dimensions and had 2,000,000 points. The data was distributed as previously but with five clusters.

Results. In Table II we have tabulated these results. True positives indicate the number of correct labels, i.e., the number of outliers that were identified as ODDs by our algorithm and the false negatives indicate the number of outliers that the algorithm failed to identify as ODDs.

For the majority of the experiments, the sensitivity was greater than 0.99. In other words, the algorithm identifies almost all outliers as ODDs. For the majority of the experiments, the specificity was greater than 0.99 and the picks included very few false negatives, i.e., those outliers that the algorithm failed to identify as ODDs.

5.3 KDD Data: Network Alert Data

We did experiments using a KDD-99 data set [<http://kdnuggets.org/>] to create several data sets to test our algorithm. Since it can be tricky to identify a point a-priori as an outlier in a data set, we used the KDD labels. The data set had a total

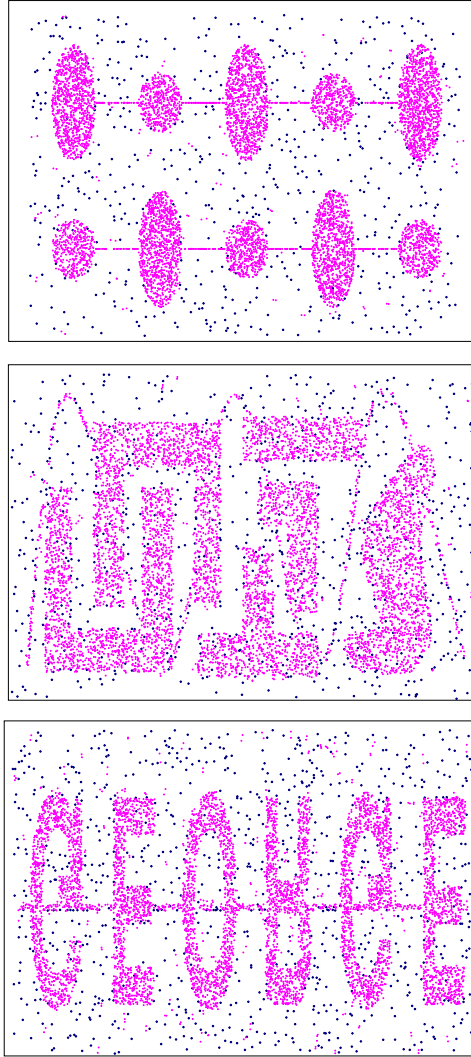


Fig. 1. Visual description of results of outlier detection for a data set drawn from Cure/Chameleon data sets

311,029 records with 37 different types of network attack patterns and records labelled “normal”. We picked one record each at random from every attack type and randomly picked either $\{1000, 5000, 10000, 25000, 50000\}$ of the normal patterns for a total of $\{1037, 5037, 10037, 25037, 50037\}$ points respectively.

We ran our algorithm on five data sets for each size (for a total of 25 data sets) and for a maximum depth, r_{max} of six to ten. We tabulate the results in Table 2. Due to lack of space, for each data set we have chosen one sample run.

Table 1. Results of finding ODDs in several large artificial data sets

Number Points	Noise %	True Positives	False Negatives	Sensitivity	Specificity	Goodness
$0.1 * 10^6$	1	924	76	0.924	0.99995	1.92395
$0.1 * 10^6$	2	1903	97	0.9515	0.99994	1.95144
$0.1 * 10^6$	3	2957	42	0.9859	0.99979	1.98569
$0.1 * 10^6$	4	3999	1	0.9997	0.99881	1.99851
$0.1 * 10^6$	5	4988	12	0.9976	0.99976	1.99736
$0.1 * 10^6$	6	5996	3	0.9994	0.99866	1.99806
$0.1 * 10^6$	7	6721	279	0.9601	0.9999	1.96
$0.1 * 10^6$	8	7988	12	0.9985	0.99949	1.99799
$0.1 * 10^6$	9	8992	8	0.9991	0.99867	1.99777
$0.1 * 10^6$	10	9977	23	0.9977	0.99981	1.99751
$1.0 * 10^6$	1	9999	1	0.9999	0.99952	1.99942
$1.0 * 10^6$	2	19991	9	0.9995	0.99763	1.99713
$1.0 * 10^6$	3	29991	9	0.9997	0.99737	1.99707
$1.0 * 10^6$	4	39930	69	0.9982	0.99988	1.99808
$1.0 * 10^6$	5	49404	596	0.9881	0.99998	1.98808
$1.0 * 10^6$	6	59887	111	0.9981	0.9999	1.998
$1.0 * 10^6$	7	69678	320	0.9954	0.99997	1.99537
$1.0 * 10^6$	8	79935	65	0.9992	0.99985	1.99905
$1.0 * 10^6$	9	89473	527	0.9941	0.99992	1.99402
$1.0 * 10^6$	10	99711	286	0.9971	0.99984	1.99694
$2.0 * 10^6$	1	19827	91	0.9954	0.9999	1.9953
$2.0 * 10^6$	2	39564	226	0.9943	0.9998	1.9941
$2.0 * 10^6$	3	56343	3657	0.9391	0.9999	1.939
$2.0 * 10^6$	4	79979	21	0.9997	0.9999	1.9996
$2.0 * 10^6$	5	99999	1	0.9999	0.9971	1.997
$2.0 * 10^6$	6	11999	1	0.9999	0.9993	1.9992
$2.0 * 10^6$	7	139044	956	0.9931	0.9999	1.993
$2.0 * 10^6$	8	158687	1313	0.9917	0.9999	1.9916
$2.0 * 10^6$	9	17998	2	0.9999	0.8731	1.873
$2.0 * 10^6$	10	19981	19	0.9999	0.9995	1.9994

We have tabulated the depth, r_{max} , number of correct outlier labels, sensitivity, specificity and goodness.

Results. In all our runs more than half the attack patterns were flagged as ODDs by our algorithm. Notice that except for one, all of the goodness values are above 1.5 and most of them are greater than 1.6.

Typically, as r_{max} is increased the number of points flagged as ODDs increases. This would mean that sensitivity, which measures true positives, goes up but false positives go up too, reducing the specificity. In Table 2, for all experiments more true outliers could have been flagged as ODDs than the number indicated, but it would have resulted in a loss of goodness. Moreover, the attack types that the algorithm failed to identify were invariably almost the same over all the different experiments.

Table 2. Results of finding ODDs in network alert data

Number Points	Depth	True Positive	Sensitivity	Specificity	Goodness
1000	6	24	0.648648649	0.94021215	1.588860799
1000	9	33	0.891891892	0.753134041	1.645025932
1000	6	22	0.594594595	0.941176471	1.535771065
1000	7	30	0.810810811	0.849566056	1.660376867
1000	6	22	0.594594595	0.947926712	1.542521306
5000	8	27	0.72972973	0.900933095	1.630662825
5000	9	28	0.756756757	0.848520945	1.605277702
5000	10	33	0.891891892	0.786182251	1.678074143
5000	7	30	0.810810811	0.922771491	1.733582302
5000	9	33	0.891891892	0.805042684	1.696934576
10000	8	25	0.675675676	0.926472053	1.602147729
10000	9	28	0.756756757	0.870279964	1.627036721
10000	9	27	0.72972973	0.89628375	1.62601348
10000	10	31	0.837837838	0.786191093	1.624028931
10000	10	28	0.756756757	0.861313141	1.618069898
25000	9	27	0.72972973	0.880257219	1.609986949
25000	8	29	0.783783784	0.90142589	1.685209673
25000	10	26	0.702702703	0.89603387	1.598736573
25000	10	27	0.72972973	0.885809003	1.615538732
25000	10	29	0.783783784	0.86995247	1.653736254
50000	10	23	0.621621622	0.87401323	1.495634852
50000	9	30	0.810810811	0.937706097	1.748516908
50000	10	23	0.621621622	0.916841537	1.538463159
50000	9	26	0.702702703	0.921737914	1.624440617
50000	10	27	0.72972973	0.893478826	1.623208555

5.4 Identifying a Minority Class

In the experiments just described, streaming ODDs were defined by using leaves in the trees that contain a single point. In the next series of experiments, we relaxed this restriction and let the leaves contain k or fewer points. This time we also picked k attack patterns for every attack type.

Experiment 1. The results are summarized in Table 3, where k is called the threshold. This time all of the goodness values are above 1.5 and again most of them are greater than 1.6.

Experiment 2. In another experiment with the network data set, we picked all the examples of those attack types that had less than 25 examples. In total there were 166 attack patterns. There were 20 such attack types. We also randomly picked 5000 points. We tried various threshold cardinalities.

The best result (in terms of goodness) was for $k = 7$ of a k -ODD, which gave goodness of 1.59 and 121 of 166 attack patterns were identified. For $k = 9$ we were able to capture 146 of 156 attack patterns, but the specificity was low.

Table 3. Results of finding those outliers in network alert data that might not occur as singletons

Points	Threshold	ODDs	True Positive	Sensitivity	Specificity	Goodness
5000	2	73	54	0.739726027	0.8928	1.632526027
5000	3	103	69	0.669902913	0.9084	1.578302913
5000	4	132	105	0.795454545	0.8156	1.611054545
5000	5	160	114	0.7125	0.909	1.6215
10000	2	73	53	0.726027397	0.8954	1.621427397
10000	3	103	77	0.747572816	0.9022	1.649772816
10000	4	132	92	0.696969697	0.903	1.599969697
10000	5	160	119	0.74375	0.8818	1.62555
25000	2	73	51	0.698630137	0.90832	1.606950137
25000	3	103	83	0.805825243	0.84108	1.646905243
25000	4	132	98	0.742424242	0.9188	1.661224242
25000	5	160	104	0.65	0.94492	1.59492

The higher cardinality works because (1) Some patterns of the same attack types are very similar. (2) Different attack types sometimes also have similar patterns. (3) Some normal patterns occur with the attack types.

5.5 NASA Shuttle Data

This data has seven classes and 14,500 trained examples in nine dimensions [http://kdnuggets.org/]. There are 11,478 examples of class one, 12 examples of class two, 38 examples of class three, 2155 examples of class four, 807 examples of class five, 4 examples of class six and 2 of class seven.

Our algorithm identifies all 6 points belonging to class six and seven as ODDs in a list of 36 ODDs.

5.6 Building Forests

Instead of using single trees to find streaming ODDs trees we can also use forests of trees to compute streaming CDDs. In Table 4 we have presented some results with one technique for creating forests. The data set consisted of 2 million points with outliers varying from 1-10%.

Results. We have tabulated the depth at which both specificity and sensitivity is greater than 0.99 for the first time (with noise of 9%, the forest approach could not reach the level of 0.99 sensitivity). It can be seen though that for most experiments the depth of a forest is less than that of a single tree.

A number of approaches can be used to create a forest. Our idea is simple:

1. Insert a new point in the tree within whose root's bounds the point falls.
2. If no such tree exists:
 - (a) If a tree has less than maximum depth, insert the point in that tree.
 - (b) Else create a new tree.

Table 4. Maximum depth needed to achieve greater than 0.99 specificity and sensitivity for streaming ODDs for a data set with 2×10^6 points and outlier varying from 1 – 10% using single tree and forest

Outlier Percent	Single Tree	Forest
1	9	7
3	10	9
4	9	7
5	7	7
6	8	8
7	10	10
8	11	9
9	7	-
10	9	7

- (c) Once the number of trees is equal to maximum number of allowable trees in the forest, insert the new point in a tree that will require the least doubling to accommodate the point within its bounds.

5.7 Discussion

We have conducted experiments with two artificial and two real life data sets.

The only variable we need to fix to run the algorithm is to fix the maximum depth of the dyadic tree.

The maximum depth of the tree determines the number of points that are identified as ODDs. More the depth, greater the number of points identified as ODDs. In our experiments a depth of 6-10 seems to have worked well. The same holds true for identifying member of a minority class.

One possible drawback of our approach is sparse data, in which it is difficult to define what an outlier is.

6 Conclusions

In this work, we have introduced a streaming algorithm for detecting outliers that is simple, effective and naturally exploits the multiscale nature of many common data sets. It is based upon a natural modification of a dyadic decomposition of a data set when the data is presented in the form of a stream and only a finite amount of space is available to construct the dyadic tree.

Once we have constructed a dyadic tree under a streaming setting we use it to find outliers in a streaming setting. We have described a few modifications to our approach, e.g., forests, k-ODDs. Finally, we have conducted experiments on artificial and real data sets to demonstrate the use of our algorithm.

References

1. Knorr, E., Ng, R.: Algorithms for mining distance based outliers in large datasets. In: VLDB '98. Proceedings of International Conference on Very Large Databases, pp. 392–402 (1998)
2. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. In: SIGMOD '00. Proceedings of the ACM International Conference Management Of Data, pp. 427–438. ACM Press, New York (2000)
3. Breunig, M.M., Kriegel, H., Ng, R.T., Sander, J.: Lof: Identifying density based local outliers. In: Proceedings of the ACM International Conference Management Of Data, pp. 93–104. ACM Press, New York (2000)
4. Jagdish, H.V., Koudas, N., Muthukrishnan, S.: Mining deviants in a time series data base. In: VLDB '99. Proceedings of International Conference on Very Large Databases (1999)
5. Muthukrishnan, S., Shah, R., Vitter, J.S.: Mining deviants in time series data streams. Technical Report DIMACS TR 2003-43, DIMACS (2003)
6. Nasraoui, O., Uribe, C.C., Coronel, C.R., Gonzalez, F.: Tecno-streams: Tracking evolving clusters in noisy data streams with a scalable immune system learning model. In: ICDM'03. Third IEEE International Conference on Data Mining, IEEE Computer Society Press, Los Alamitos (2003)
7. Sheikholeslami, G., Chatterjee, S., Zhang, A.: Wavecluster: A multi-resolution clustering approach to very large databases. In: VLDB '98. Proceedings of the 24th VLDB Conference (1998)
8. Chakravarty, S.V., Ghosh, J.: Scale based clustering using the radial basis function network. IEEE Transactions on Neural Networks (1996)
9. Wong, Y.F.: Clustering data by melting. Neural Computation 5(1), 89–104 (1993)
10. Leung, Y., Zhang, J.S., Xu, Z.B.: Clustering by scale-space filtering. IEEE Transactions on Pattern Analysis And Machine Intelligence 22(12) (2000)
11. Roberts, J.S.: Parametric and non-parametric unsupervised cluster analysis. Pattern Recognition 30(2), 261–272 (1997)
12. Barbara, D., Chef, P.: Using fractal dimension to cluster data sets. In: Proceedings of the 6th ACM SIGKDD, pp. 260–264. ACM Press, New York (1999)
13. Warnekar, C.S., Krishna, G.: A heuristic clustering algorithm using union of overlapping pattern cells. Pattern Recognition 11, 85–93 (1979)
14. Schikuta, E., Erhart, M.: The bang clustering system: A grid based data analysis. In: Proceedings Advances in Intelligent Data Analysis, Reasoning About Data 2nd International Symposium, pp. 513–524 (1997)
15. Schikuta, E.: Grid clustering: A fast hierarchical clustering method for very large data sets. In: Proceedings 13th International Conference on Pattern Recognition, vol. 2, pp. 101–105 (1996)
16. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second Intl Conference on Knowledge Discovery and Data Mining (1996)
17. Guha, S., Rastogi, R., Shim, K.: Cure: An efficient clustering algorithm for large databases. In: Proceedings of 1998 ACM-SIGMOD International Conference on Management of Data, ACM Press, New York (1998)
18. Karypis, G., Han, E.H., Kumar, V.: Chameleon: Hierarchical clustering using dynamic modelling. IEEE Computer 32(8), 68–75 (1999)
19. Zhang, T., Ramkrishnan, R., Linvy, M.: Birch: An efficient data clustering method for very large databases. SIGMOD 25(2), 103–114 (1996)

20. Ng, R., Han, J.: Very large data bases. In: VLDB '94. Proceedings of the 20th International Conference on Very Large Data Bases, pp. 144–155 (1994)
21. Guha, S., Mishra, N., Motwani, R., O'Callaghan, L.: Clustering data streams. In: The Annual Symposium on Foundations of Computer Science, IEEE, Los Alamitos (2000)
22. Bradley, P.S., Fayyad, U.M., Reina, C.A.: Scaling clustering algorithms to large databases. In: Terano, T., Chen, A.L.P. (eds.) PAKDD 2000. LNCS, vol. 1805, pp. 9–15. Springer, Heidelberg (2000)
23. Farnstrom, F., Lewis, J., Elkan, C.: True scalability of clustering algorithms. SIGKDD Explorations (2000)
24. Aggarwal, C.C., Han, J., Yu, P.S.: A framework for projected clustering of high dimensional data streams. In: Proceedings of the 30th VLDB Conference (2004)
25. Stein, E.M.: Singular Integrals and Differentiability Properties of Functions. Princeton University Press, Princeton (1970)

VISRED – Numerical Data Mining with Linear and Nonlinear Techniques

Antonio Dourado, Edgar Ferreira, and Paulo Barbeiro

Centro de Informática e Sistemas

Department of Informatics Engineering, University of Coimbra, 3030-290 Coimbra Portugal
{dourado, edgar, pbarbeiro}@dei.uc.pt

Abstract. Numerical data mining is a task for which several techniques have been developed that can provide a quick insight into a practical problem, if an easy to use common software platform is available. **VISRED-** Data Visualisation by Space **Reduction** presented here, aims to be such a tool for data classification and clustering. It allows the quick application of Principal Component Analysis, Nonlinear Principal Component Analysis, Multi-dimensional Scaling (classical and non classical). For clustering several techniques have been included: hierarchical, k-means, subtractive, fuzzy k-means, SOM- Self Organizing Map (batch and recursive versions). It reads from and writes to Excel sheets. Its utility is shown with two applications: the visbreaker process part of an oil refinery and the UCI benchmark problem of breast cancer diagnosis.

Keywords: multidimensional scaling; numerical data mining; principal component analysis; applications.

1 Introduction

Monitoring of large scale industrial processes, diagnosis in medical problems, decision support in finance and services, are tasks that can profit from data-mining of the numerical data available in today's information system. In industry every day a huge amount of data, from thousands of sensors, is available and should be used for supporting mill's managers and operators. Medical doctors have databases with thousands of patients that can be precious in supporting diagnosis. An application to ease that operation is being developed. It is based on the concept of reduction of the dimension of the original space to a three or two dimensional space, where information is easily represented and interpreted by humans [1]. Multidimensional scaling [2] has been adopted for that purpose. It is a technique that has been used in social sciences for long-time, rooted in the pioneer works of psychologists Young and Householder [3]. Its application to industrial problems has been recently identified as an important development in industrial data mining activities [4][5][6]. Presently there are some efforts to use it in process control as the present work does. In medical diagnosis, when a large amount of quantitative data from a population of patients is available, multidimensional scaling may give a quick and important support to diagnosis.

VISRED- Data Visualization by Space Reduction is an application developed in the Matlab environment[22] to adapt and integrate a set of techniques for quantitative (numerical) data mining. It integrates the following techniques:

- Linear Principal Component Analysis
- Non Linear Principal Component Analysis
- Classical Multidimensional Scaling
- Multidimensional Scaling

After reduction of the dimension, data is clustered (in the reduced space) by one of several clustering techniques available (hierarchical, k-means, fuzzy k-means, subtractive, dignet, SOM, RSOM).

Several measures can be chosen in the original high dimensional space and in the target low dimensional space for the dissimilarities and to optimize the results.

In the following paragraphs the application will be presented. In paragraph 2 a brief presentation of the techniques intends to support the understanding of the user-interface presented in paragraph 3 (for dimension reduction) and 4 (for clustering). Its application to two representative cases will be shown in paragraph 5, followed by the conclusions.

2 A Brief Description of the Techniques for Dimension Reduction

2.1 Linear Principal Component Analysis

Principal Components Analysis (PCA) [7] is a technique for simplifying a high-dimensional data set by reducing it to a lower dimension. PCA transforms the data, using an orthogonal linear transformation, to a new coordinate system, such that the greatest variance, by any projection of the data, lies on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. PCA can be used for dimensionality reduction in a data set while retaining those characteristics of the data that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. The low-order components must contain the main features of the data to preserve information. However this is not always possible, and is not whenever the data is produced by a nonlinear system.

2.2 Nonlinear Principal Component Analysis

Nonlinear Principal Component Analysis allows extending to nonlinear relations the concept of principal component, which is the decisive concept for data representation in a reduced space. The technique implemented in VISRED is the one of Hsieh [8] and based on the pioneer developments of Kramer [9] with the contributions of Daszykowski [10].

Nonlinear PCA is implemented by using a bottleneck neural network (BNN), a neural network which has few neurons (normally one to three) in the central layer, surrounded by a symmetric architecture of hidden, input and output neurons (see[9]).

The number of inputs is the same of the outputs, and the goal of training is to achieve as output the same values that were provided as inputs. The hidden neurons are nonlinear. The bottleneck layer will contain a representation of data with fewer dimensions, nonlinearly related to the inputs. There are two distinct ways to obtain several nonlinear (principal) components. If n nonlinear components are desired, one of the approaches would be to use n neurons in the bottleneck layer, to obtain n nonlinear components.

A second approach consists in using only one neuron in the bottleneck layer to extract the first nonlinear principal component. Then, the residual error is used as input (and target output) for a second BNN that computes the second nonlinear principal component, and so on for higher order components. This approach, implemented in VISRED, assures that the nonlinear principal components are orthogonal to each other, contrarily to the first approach [9]. The implementation is based on the software NeuMATSA from Hsieh [8].

2.3 Classical Multidimensional Scaling and Multidimensional Scaling

Given a set of p points in an original n dimensional space, some measure of dissimilarity between each pair of them can be defined. A dissimilarity matrix is then constructed with these dissimilarities. Using Euclidean distances as the dissimilarity measure, construct the dissimilarity matrix Δ_n in the original n -dimensional space. Now classical multidimensional scaling will find a distribution of points into an m -dimensional space, $m \ll n$, such that the Euclidian distances between the dissimilarity matrix Δ_n and the dissimilarity matrix Δ_m in the m -dimensional space is minimized in the least squares sense (1). The matrix distance (1) is defined as the sum of the Euclidian distances between every point in one and the corresponding point in the other. For more details see [6].

$$J = \|\Delta_n - \Delta_m\|^2 = \sum_{i=1}^p \sum_{j=1}^p \left\| (\Delta_n)_{ij} - (\Delta_m)_{ij} \right\|^2 \quad (1)$$

If the distance used to quantify dissimilarity is not Euclidian, but for example, City block, Mahalanobis, etc., then one must apply multidimensional scaling. Multidimensional scaling is an optimization process aiming at minimizing a distance between the two dissimilarity matrices. If the final distance would be zero then the points in the reduced space would be a perfect view of the points in the original high dimensional space. In this situation all the information content expressed by the positions of the points is perfectly preserved when the dimension reduction is performed. This may be said a topology preserving method. However in practical problems that distance J is never zero and its minimization is the goal. The minimization is performed by some optimization technique. The optimization must be initialized at some matrix. Usually one chooses to initialize by classical multidimensional scaling that can be computed by matrix calculus and produces the best lower (m)-rank approximation in the least-squares sense (1). This means that classical scaling is equivalent to multidimensional scaling with Euclidian distance.

3 The User Interface of VISRED

Figure 1 shows the main interface of VISRED. It is composed by three main panels: The Raw Data panel, the Dimension Reduction panel, and the Clustering panel. It has been conceived to ease the task of defining the needed parameters for the several methods.

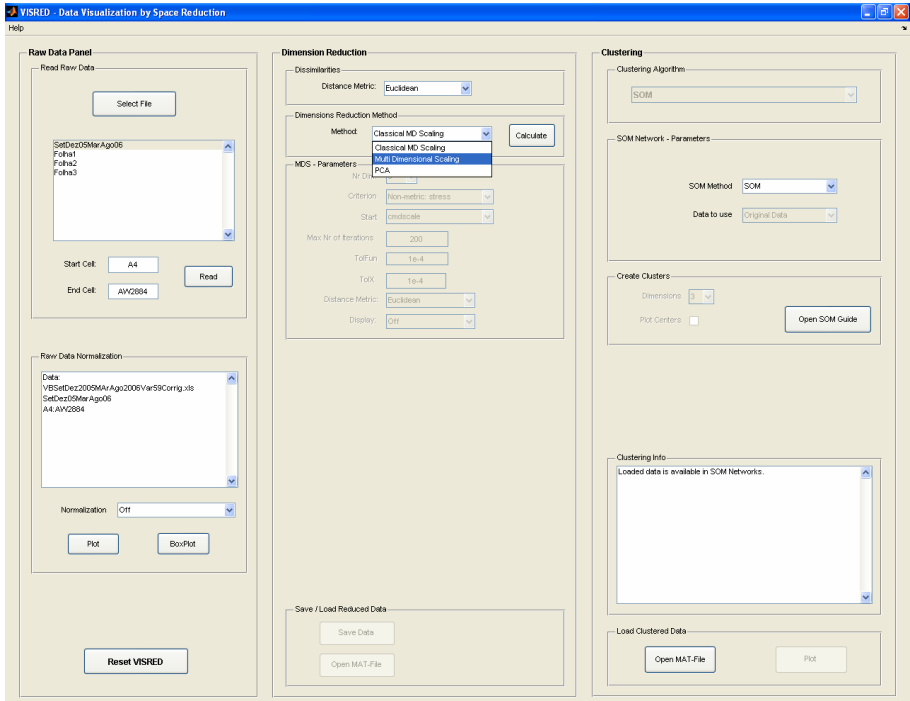


Fig. 1. The main user interface of VISRED. It is divided in three main parts: Raw Data Panel, Dimension Reduction, and Clustering.

3.1 The Raw Data Panel: Reading, Normalizing, Plotting

The user selects an excel file and a sheet in the file where the data is collected. The first columns of the sheet can have non-numerical data for labeling of the points. The remaining columns contain the coordinates of the first dimension and the following columns the following dimensions. There is no limit, except memory and computational times, for the number of dimensions. The user must specify the first and the last cells to be read. The first cell is A4 by default. It is the one that contains the names of the labels of the points (it can be empty). The last one is the last data to be used.

After reading the data, normalization is applied or not. Data can be normalized by mean value only, by standard deviation only (if it is greater than one), or by both simultaneously.

Data can then be plotted in a new window and analyzed (before or after normalization) and outliers can be observed. User can decide to eliminate some outliers in the excel sheet and restart the analysis.

3.2 Dimension Reduction

The data is now ready for dimension reduction. Here methods implemented in the Matlab Statistics Toolbox are extensively used though the interface. Firstly one must choose one of the methods: PCA (Principal Component Analysis), Classical Multidimensional Scaling, Multidimensional Scaling. The used distance is chosen from the following possibilities: Euclidian, standardized Euclidian, Mahalanobis, city block (Manhattan), Minkowski, cosine, correlation, Chebychev, Hamming, Jaccard.

3.2.1 Multidimensional Scaling

The user has several parameters to control the optimization process of multidimensional scaling (number of iteration, stopping criteria, output information)

MDS - Parameters	
Nr Dim.	3
Criterion	Non-metric: stress
Start	cmdscale
Max Nr of iterations	200
TolFun	1e-4
TolX	1e-4
Distance Metric	Euclidean
Display	Off

Fig. 2. Parameters that can be chosen when applying Multi Dimensional Scaling

Multidimensional scaling may be numeric, where numeric distances are considered for the objective function to be minimized, or non-numeric where only ordinal relations must be preserved. In VISRED, a slightly improved version of Matlab `mdscale m`-function is used: one can choose which distance will be used to calculate the dissimilarity matrix, instead of being restricted to the standard metric (Euclidean). The used distance is selected in the user interface.

After applying Classical Multidimensional Scaling or Principal Component Analysis, it is possible to analyze the data based on the eigenvalues. The pareto analysis (Fig.3) allows the user to compute the number of dimensions needed for a certain percentage of explained variation. The user can also compute and graph in a separate window the distances between the elements of the dissimilarity matrices and analyze the dispersion around the average (Fig. 4) or to count the outliers for a certain percentage. Outliers influence can be studied here and identified. Fixed a percentage of outliers, VISRED identifies in a separate window the data points that result. The user may chose to eliminate them in the data sheet and restart the study.

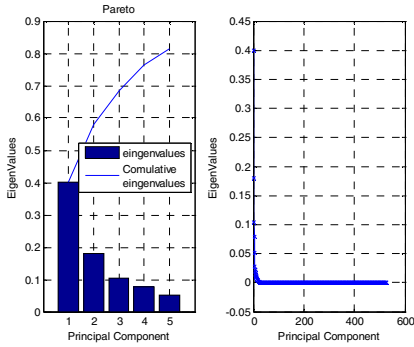


Fig. 3. Pareto Analysis

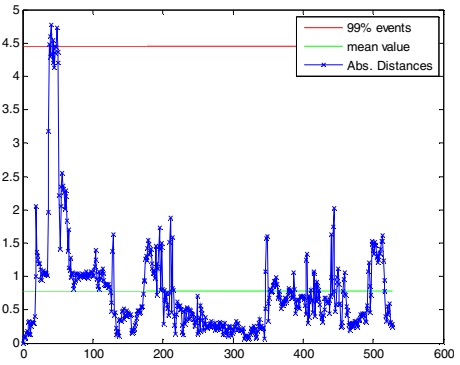


Fig. 4. Absolute Distance Analysis

3.2.2 Nonlinear PCA

In order to simplify user’s manipulation of values and training of the BNN network, a user-friendly interface was built (Fig.5), where one can easily adjust the needed parameters of the BNN method. Up to three nonlinear principal components can be successively calculated and saved. Points and labels can also be saved for later clustering on VISRED interface.

The nonlinear principal components define a nonlinear base where the data can be represented. Fig. 6 shows, for an example, three components and the resulting data representation in the 3-dimensional space, to which the clustering operation will then be applied.

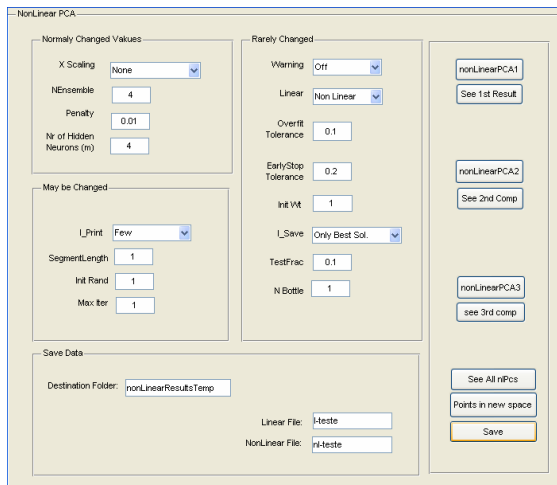


Fig. 5. Nonlinear PCA interface

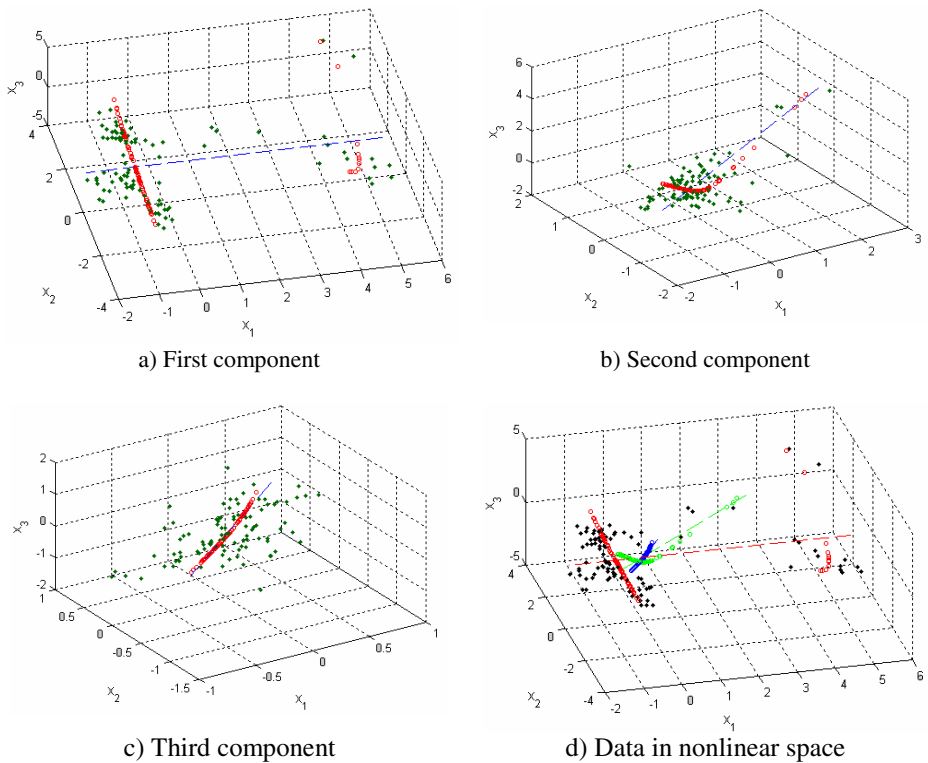


Fig. 6. The nonlinear principal components extraction process. The process starts on input data until all nonlinear principal components are extracted and points are represented in 3D space with these components.

4 Clustering in the Reduced Space

The data obtained from dimension reduction can be saved in a mat file for later processing or can be used immediately for clustering. The following clustering methods are available: hierarchical, k-means, subtractive, fuzzy c-means, dignet, SOM-Self Organizing Map, Recursive SOM-Self Organizing Map. Classical clustering methods like hierarchical, k-means, subtractive, fuzzy c-means are fully implemented and described by Statistics and Fuzzy Logic Toolboxes of Matlab. For each method its optional parameters can be chosen in the user interface (number of clusters, distance metrics to be used between points, etc).

Dignet is a self-organizing neural network that can store and classify noisy inputs without supervised training [11] [12]. Its self-organization capability is based on the idea of competitive generation and elimination of attraction wells. Each well is characterized by its center, width (threshold), and depth. The wells are generated around presented patterns which are clustered according to their distance from the center of wells. The center of a well is moved dynamically towards the highest

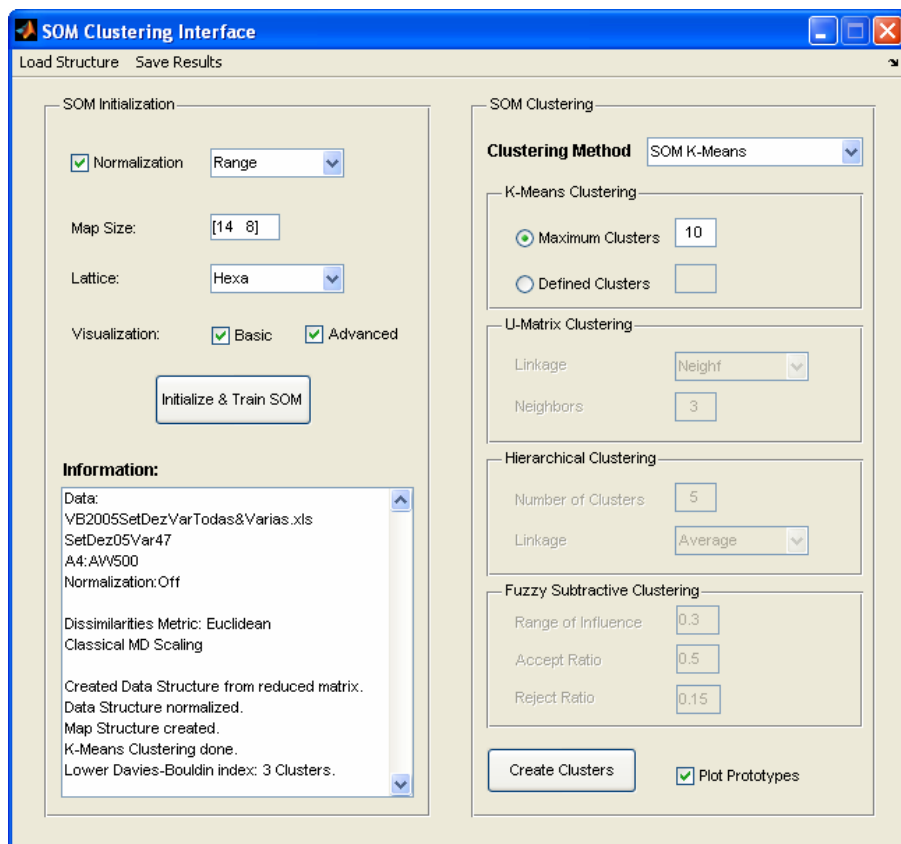


Fig. 7. SOM Clustering implementation in VISRED

concentration of clustered points in the pattern constellation. The depth of a well indicates the strength of learning, and influences the inertia of the center of the well when new data falls within its region of attraction; the deeper the well is, the less its center moves towards a new data point [12].

The similarity between patterns in Dignet is measured using a distance which could be Euclidean, angular (cosine) or hyper-cubic (Chebychev). For angular metric it is assumed that all patterns are normalized [11], so that the magnitude of a pattern does not affect the classification capability of the network.

Self-Organizing Map (SOM), proposed by Kohonen [13], is an unsupervised neural network method which has properties of both vector quantization and vector projection algorithms. The prototype vectors are positioned on a regular low-dimensional grid in an ordered fashion, making the SOM a powerful visualization tool [14]. Each neuron is a n -dimensional weight vector where n is equal to the dimension of the input vectors. The neurons are connected to adjacent neurons by a neighborhood relation, which dictates the topology, or structure, of the map [15]. In the approach, topology is defined by local lattice structure (hexagonal or rectangular).

The SOM can be thought of as a net which is spread to the data cloud. The SOM training algorithm moves the weight vectors so that they span across the data cloud and so that the map is organized such that neighbor neurons on the grid get similar weight vectors.

After weight vectors adjustment clustering techniques are applied to them. Each original vector gets the cluster index that is attributed to its nearest weight vector in the space distribution. A developed user interface helping to configure its application, in presented in Fig.7. It can be applied to high and low dimensional data.

SOM Clustering Interface is composed by two main sections (Fig.7): initialization (left panel) and clustering (right panel). On the left panel, it is possible to control several SOM training parameters; on the right panel there are four clustering methods available. The k-means chooses the number of clusters by the lowest Davies-Bouldin index; U-matrix clustering uses the matrix of distances between weight vectors.

This interface works over the SOM Toolbox Version 2 of [14] [15]. Figure 8 shows a typical 2 dimensions result.

The recursive SOM network is a generalization of SOM that learns to represent sequences recursively. Its resulting representations are adapted to the temporal statistics of the input series [16].

The SOM Structured Data network is also an extension of the standard SOM model, which allows the mapping of structured objects into a topological map. This mapping can then be used to discover similarities among the input objects [17].

Implementation of these networks is based on Recurrent Self-Organizing Map (RSOM) Toolbox for MATLAB, developed by A. R. Saffari and A. Alamdari [18].

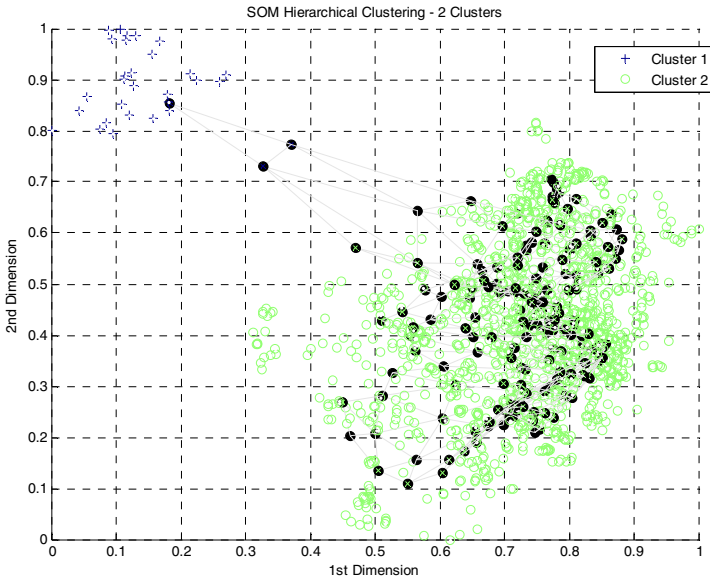


Fig. 8. SOM Hierarchical Clustering. Data has two isolated clouds of points; the linked weight vectors are the result of SOM training and are positioned in order to represent distribution and density of data. Hierarchical clustering applied to weight vectors perfectly separates the point clouds of original data.

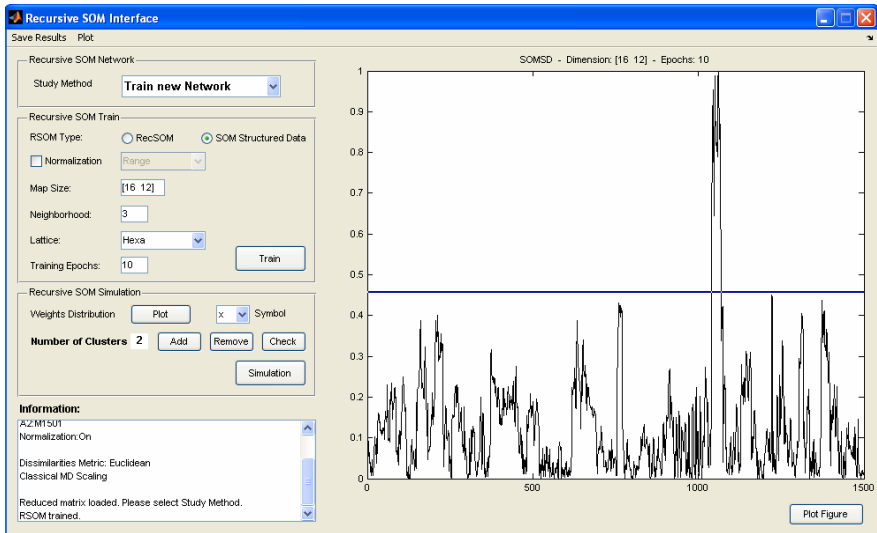


Fig. 9. Recursive SOM Clustering implementation in VISRED. This method can be applied to both high and low dimensional data.

Recursive SOM Interface allows training of recursive SOM with defined parameters, like network type (recursive or structured data); the other parameters are similar to those used in SOM training. Graphic area shows network output for training data; this graphics can be divided into clusters by using mouse movable lines that represent cluster thresholds.

Results of recursive SOM train and clustering can be saved and applied to unknown data that represents a similar dimensional space (dimensions of train and simulation must match).

5 Applications

5.1 The Visbreaker Process

The Visbreaker Unit, an important process in an oil refinery, is intended to reduce the viscosity of some intermediate products (the residual coming from the vacuum column) in the refining chain. With this objective a thermal cracking process is used with a relatively low temperature, and a long residence time. As a result of the thermal process a low viscosity visbreaker residual is obtained, as well as lighter products, such as hydrocarbonates (gas oil diesel, gasoline and gases). The great economical advantage of the visbreaker process lies in fact that it produces a residual with a lower viscosity than the load feed. By this way, it is possible to use a lower quantity of “cutterstocks” (some of them of high benefit) for the production of fuel oil.

Figure 10 [19] presents its flow sheet. It is composed of several sub-processes, and its main part is a kiln operating at about 310 °C. Actually, data from 160 tags is available. From these, after correlation analysis and process expertise, 59 were selected as sufficiently representative of the process. Multidimensional scaling is then applied to those 59 dimensions.

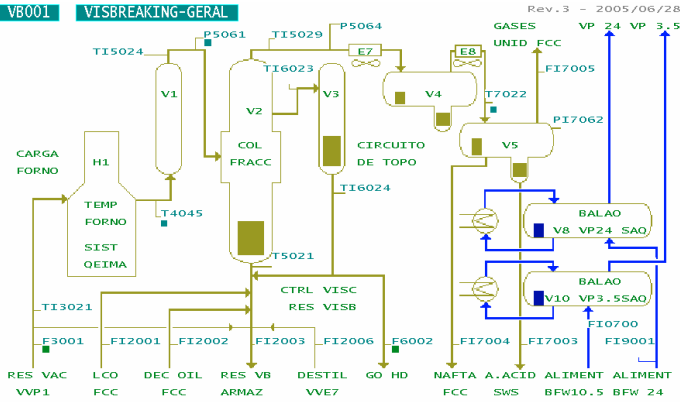


Fig. 10. Schematic representation of the visbreaker process

The data from March to August 2006 is averaged every hour and considered for the VISRED.

Fig. 11 shows the results with Classical Multidimensional Scaling (CMDS). Two regions and some outliers are identified. These regions correspond to different operational conditions.

Applying Multidimensional Scaling, with City block distance, Figure 15 is obtained. The dispersion of the points is very similar to CMDS (see however the difference in axes scales).

There are some points in Figures 11 and 12 that could be considered outliers (although they are an average of one hour). Proceeding by this way, applying again CMDS with Euclidian distance, followed by hierarchical clustering with Euclidian averaged distance, with 5 clusters, one obtains Fig. 13. Now two main clouds are identified and grouped in one cluster for each. With Matlab graphical capabilities one can see the dates corresponding to each of the clusters, which can help the plant managers to evaluate the performance of the mill.

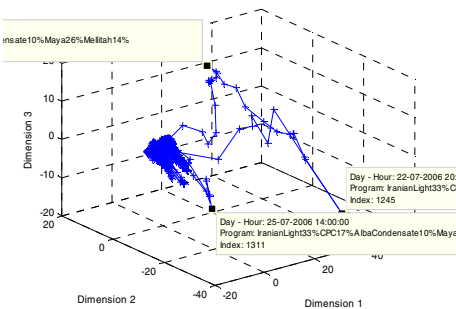


Fig 11. Classical scaling with Euclidian distance

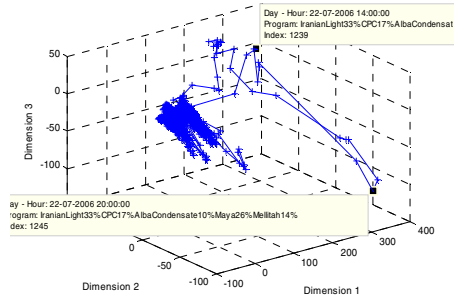


Fig. 12. Multidimensional scaling with city-block distance

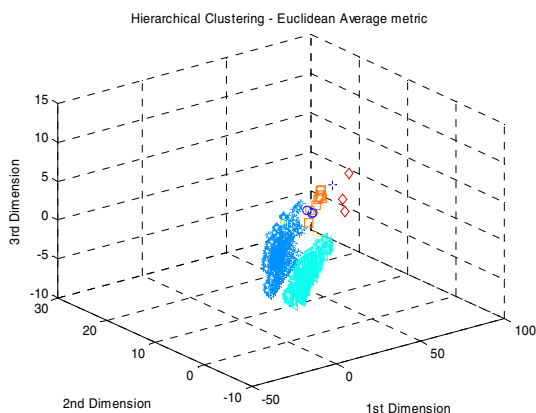


Fig. 13. Hierarchical clustering of CMDS results after elimination of some outliers of Fig. 12. The changement of axes is due to the graphication part.

5.2 Breast Cancer Diagnosis

Breast cancer diagnosis (1- malign or 2-benign) is performed using 30 continuous variables measured in 567 patients , (UCI Repository of Machine Learning Databases and Domain Theories [20]) from the University of Wisconsin Hospitals, Madison [21]. Using VISRED the data has been reduced to three dimensions. Table 1 synthesizes the results for several methods implemented in VISRED.

Table 1. Breast cancer diagnosis using several combinations of reduction and clustering in VISRED. Best predicting combinations are marked.

Normali- zation	Red. Method (Criterium /Start)	Distance Metric	Clustering Method	Dim .	Clustering Metric	Err ors	Predicting %
Off	CMD	Euclidean	Hierarchical	3	Euclidean	375	33,86
Standard	CMD	Euclidean	Hierarchical	3	Euclidean	106	81,31
On	CMD	Euclidean	Dignet	3	Euclidean	38	93,30
Range(SOM)			SOM / K-Means	30		25	95,59
Range (SOM)			SOM / Subtract	30		33	94,18
On	CMD	Chebychev	Dignet	3	Euclidean	66	88,36
On	CMD	Cityblock	Dignet	3	Euclidean	37	93,47
On	MDS /NM stress	Cityblock	Dignet	3	Euclidean	36	93,65
On	NL-PCA	Euclidean	C-Means	3		7 8	86,24
On	NL-PCA	Euclidean	Subtractive	3		7 5	86,77
On	NL-PCA	Euclidean	SOM/ K-Means	3		8 2	85,54
Range	RecSOM			30		7 2	87,30
Range	SOM-SD			30		6 7	88,18

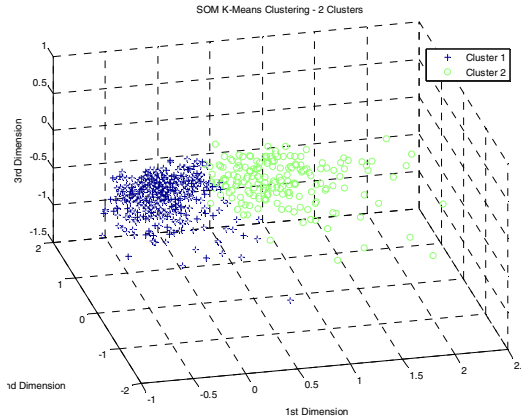


Fig. 14. Space distribution by PCA projection and clustering plot of the best breast cancer predicting combination using SOM and k-means

As can be seen in Table 1, several combinations of methods achieve a good match ratio on predicting breast cancer diagnosis. The best combination uses k-means clustering after SOM training with no dimensional reduction; the second best performance is achieved by dignet clustering with Euclidean metric, after dimensional reduction by MDS cityblock.

Both predicting rates (95,59% and 93,65%) show that good combinations of data-mining methods can be applied to real data. VISRED has a high flexibility in choosing the methods to be applied.

6 Conclusion

VISRED is an application to ease the activity of researchers and practitioners of numeric data mining. It is public, under GNU principles, and the main effort is being put into the design of friendly and useful user interfaces, in Matlab environment [22]. It is a work in progress, but its present form allows already the quick study of any application with data in standard excel sheets. Comparison of several techniques, linear or non linear, for data analysis based on dimensional reduction can readily be done and the extensive graphical tools present allow the user to gain an insight into the problem under study.

The authors hope that this platform will be useful for all the data mining community.

The software is public GNU licensed and is available at <http://eden.dei.uc.pt/~dourado/Visred/VisRed.zip>

Further work to improve the VISRED platform is being developed to enhance the multidimensional scaling performance. The optimization task should avoid local minima and find the global optimal dissimilarity matrix in the reduced space. Meta heuristics like genetic algorithms and simulated annealing are being investigated with this aim.

The authors would appreciate any comments, suggestions, and contributions of the data mining community in order to build up a powerful free software for data mining.

Acknowledgments

This work was supported by Portuguese Foundation for Science and Technology (Project CLASSE, POSC/EIA/58162/2004) and Feder. Edgar Ferreira and Paulo Barbeiro were supported by a BIC grant from CLASSE project. The Sines refinery data was supplied by Eng. Dora Nogueira and Eng. Luís Amaral. The authors would like to thank the authors of GNU software that have been adapted to be used in VISRED (W.W. Hsieh, J. Vesanto, A. R. Saffari).

References

1. de Oliveira, M.C.F., Levkowitz, H.: From visual data exploration to visual data mining: A survey. *IEEE Trans on Visualization and Computer Graphics* 9(3), 378–394 (2003)
2. Borg, I., Groenen, P.: *Modern Multidimensional Scaling, Theory and Applications*, 2nd edn. Springer, Heidelberg (2005)
3. Young, G., Householder, A.S.: A note on multidimensional psycho-physical analysis. *Psychometrika* 6, 331–333 (1941)
4. Cox, T.F.: *Multidimensional Scaling in Process Control*. In: Khattree, R. (ed.) *Handbook of Statistics 22. Statistics in Industry* North Holland (2003)
5. Cox, F.T.: *Multidimensional Scaling*, 2nd edn. Chapman and Hall CRC Press, New York (2001)
6. Matheus, J., Dourado, A., Henriques, J.: *Iterative Multidimensional Scaling for Industrial Process Monitoring*. In: *Trans. IEEE SMC Int. Conf.*, Seoul, Taiwan, October 8-7,11, 2006 (2006)
7. Jolliffe, I.T.: *Principal Component Analysis*, 2nd edn. Springer, Heidelberg (October 2002)
8. Hsieh, W.W.: *Neuralnets for Multivariable and Time Series Analysis (NeuMATSA): A User Manual*, www.ocgy.ubc.ca/william/Pubs/NN.manual.pdf
9. Kramer, M.A.: *Nonlinear Principal Component Analysis Using Autoassociative Neural Networks*. *AIChE Journal* 37(2), 233–243 (1991)
10. Daszykowski, B., Walczak, I., Massart, D.L.: *A journey into low-dimensional spaces with autoassociative neural networks*. In: *Talanta*, vol. 59, pp. 1095–1105. Elsevier, Amsterdam (2003)
11. Thomopoulos, S.C.A., Bougoulas, D.K, Wann, C.-D.: *Dignet an Unsupervised-Learning Clustering Algorithm for Clustering and Data Fusion*. *IEEE Trans On Aerospace and Electronic Systems* 31(1) (1995)
12. Wann, C.-D., Thomopoulos, S.C.A.: *A comparative study of self-organizing clustering algorithms Dignet and ART2*. *Neural Networks* 10(4), 737 (1997)
13. Kohonen, T.: *Self-Organizing Maps*, 3rd Extended edn. Springer Series in Information Sciences, vol. 30. Springer, Heidelberg (2001)
14. Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J.: *SOM Toolbox for Matlab 5*. Helsinki University of Technology (2000)
15. Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J.: *Self-organizing map in Matlab: the SOM Toolbox*. In: *Proceedings of the Matlab DSP Conference*, Espoo, Finland, November 16-17, 1999, pp. 35–40 (1999)

16. Voegtlin, T.: Recursive self-organizing maps. *Neural Networks* 15, 979–991 (2002)
17. Hagenbuchner, M., Sperduti, A., Tsoi, A.C.: A self-organizing map for adaptive processing of structured data. *IEEE Transactions on Neural Networks* 14(3), 491–505 (2003)
18. Saffari, A.R., Alamdari, A.: Recurrent Self-Organizing Map (RSOM) Toolbox for MATLAB (July 2005), <http://www.ymer.org/amir/software/recurrent-self-organizing-maps>
19. Galp data, Sines Refinery (2006)
20. Murphy, P.M.: UCI Repository of Machine Learning Databases and Domain Theories (online), Available at <http://www.ics.uci.edu/mlearn/MLRepository.html>
21. Wolberg, W.H., Street, W.N., Mangasarian, O.L.: Machine learning techniques to diagnose breast cancer from fine-needle aspirates. *Cancer Letters* 77, 163–171 (1994)
22. Matlab is a trademark of Mathworks, Inc

Clustering by Random Projections

Thierry Urruty¹, Chabane Djeraba¹, and Dan A. Simovici²

¹ LIFL-UMR CNRS 8022, Laboratoire d'Informatique Fondamentale de Lille,
Université de Lille 1, France

urruty,djeraba@lifl.fr

² University of Massachusetts Boston, Department of Computer Science, Boston,
Massachusetts 02125, USA

dsim@cs.umb.edu

Abstract. Clustering algorithms for multidimensional numerical data must overcome special difficulties due to the irregularities of data distribution. We present a clustering algorithm for numerical data that combines ideas from random projection techniques and density-based clustering. The algorithm consists of two phases: the first phase that entails the use of random projections to detect clusters, and the second phase that consists of certain post-processing techniques of clusters obtained by several random projections. Experiments were performed on synthetic data consisting of randomly-generated points in \mathbb{R}^n , synthetic images containing colored regions randomly distributed, and, finally, real images. Our results suggest the potential of our algorithm for image segmentation.

1 Introduction

Clustering is a central preoccupation in data mining and clustering algorithms impact a multitude of data mining applications ([5,19,7]), including multimedia data mining. The problem has been studied by several research communities ranging from statistics to machine learning and the state of the art is exposed in surveys that appeared with some regularity over the years (see [12,9]). Clustering in spaces with low dimensionality is relatively easy. For example, in a unidimensional space it is easy to identify the regions of high density of points by a simple linear scan. With increased dimensionality the problem grows in complexity. The notion of projected clustering was introduced by Agrawal et al. in [1], who made the crucial observations that points may cluster better in subspaces of lower dimensionality than in the entire space \mathbb{R}^n . They developed the CLIQUE algorithm that works starting with low dimensional subspaces towards higher dimensional subspaces. In [3] Aggarwal et al. focus on a technique to discover clusters in small dimensional subspaces, which is the focus of their PROCLUS algorithm. The theoretical support of these techniques can be found in Johnson-Lindenstrauss Lemma [11] which asserts that a set of points in a high-dimensional Euclidean space can be projected into a low-dimensional Euclidean space such that the distance between any two points changes by only a factor of $1 \pm \epsilon$ for $\epsilon \in (0, 1)$. Simplifications of the proof of this result have been obtained by Frankl and Maehara [8] and by Dasgupta and Gupta [6]. An especially useful source is the monograph [17].

The number of clusters is a given parameter in PROCLUS and the algorithm identifies these clusters and a set of dimensions associated with each cluster such that the

points of the cluster are correlated with these dimensions. Another contribution to projective clustering is [2], where an objective function is introduced that takes into account a tradeoff between the dimension of a subspace and the clustering error; an extension of k -means to projective clustering in arbitrary subspaces is introduced. Our approach is similar to the approach adopted in [1] in that we construct clusters in low dimensional spaces and then select those dimensions that can best help to identify clusters in the original data set. Our main contribution consists in choosing a random frame of reference for the data set and execute the projections on the subspaces that correspond to this randomly chosen axes. We show that this process has a certain advantage over using the natural system of coordinates in that it diminishes the chance of the occultation phenomenon, which occurs when the projections of two distinct clusters of the data on a subspace are not disjoint. Static segmentation of images regarded as partitioning an image into a number of regions that represent a meaningful part of the image can be helped, as we show, by applying clustering techniques (see [10]). Our clustering algorithm combines ideas from random projection techniques and density-based clustering. The distance between points in \mathbb{R}^n is the Euclidean distance. The proposed algorithm is applicable to numeric data, that is, to data in \mathbb{R}^n and involves projecting the data on a randomly chosen base. Then, histograms of the uni-dimensional projections are combined to yield the locations of clusters in \mathbb{R}^n .

The paper begins with a probabilistic evaluation of the projection technique. Namely, in Section 2 we evaluate the probability that the distance between random projections on subspaces reproduces to a certain extent the distance between the original points in \mathbb{R}^n and the probabilities that random projections of separate clusters may have non-empty intersection and, therefore, reduce the usefulness of certain projections. In Section 3 we discuss the clustering algorithm including two important post-processing techniques and we show that the time complexity is of the order of $O(N \log N)$ when the size of the data set is large compared to the number of dimensions, comparable with density-based clustering [15]. Section 4 presents our experimental work performed on three types of data: synthetic data, data obtained from synthetic images, and data obtained from real images. Experiments with groupings of pixels extracted from images, particularly from real images show the potential of the algorithm as a segmentation technique and provide a good criterion for validation of clusterings.

2 Clusters and Random Projections

Let S be a finite subset of \mathbb{R}^n and let δ be a positive real number. Consider a measure $m : \mathcal{P}(\mathbb{R}^n) \rightarrow \mathbb{R}_{\geq 0}$. The value $m(C)$ is, in general, the volume of the projection of C on a subspace of \mathbb{R}^n .

A δ -clustering of S is a family $\kappa = \{C_1, \dots, C_p\}$ of non-empty subsets of \mathbb{R}^n (referred to as the *constituents* of the clustering) that satisfy the following conditions:

1. the sets of κ that are pairwise disjoint;
2. for every i , $1 \leq i \leq p$ density of the points of S in any of the sets C_i exceeds δ , that is, we have:

$$\frac{|S \cap C_i|}{m(C_i)} \geq \delta.$$

The *clusters* of the clustering κ are the sets $S \cap C_i$ for $1 \leq i \leq p$.

The set of points located outside the sets C_i , $\text{UNC}(\kappa) = S - \bigcup_{i=1}^p C_i$ is the *set of unclassified points of S*.

A *clustering of S* is a family $\kappa = \{C_1, \dots, C_p\}$ that is a δ -clustering of S for some $\delta > 0$.

The second condition of the above definition insures that the density of the points in each of the sets C_i is sufficiently high.

Projections on the subspace of \mathbb{R}^n determined by the coordinates i_1, \dots, i_t are denoted by $\text{proj}_{i_1 \dots i_t} : \mathbb{R}^n \longrightarrow \mathbb{R}^t$.

Let \mathbf{H} be a random $n \times n$ -matrix that is orthogonal. Such a matrix can be obtained, for example, by randomly choosing the components on an $n \times n$ -matrix using an uniform distribution on an interval and, then, applying the Gram-Schmidt technique to produce an orthogonal matrix.

A random projection of \mathbb{R}^n is a linear transformation $\Phi_{\mathbf{H}} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ defined by $\Phi_{\mathbf{H}}(\mathbf{x}) = \mathbf{H}\mathbf{x}$ for $\mathbf{x} \in \mathbb{R}^n$. The set of rows $\mathbf{u}_1, \dots, \mathbf{u}_n$ of \mathbf{H} is referred to an n -dimensional *random frame*.

Identifying clusters in one dimension is a relatively straightforward process using an algorithm (described in Section 3) that builds histograms of the line coordinates of the projections of the points. The inverse statement does not hold; if the projection of a subset K of \mathbb{R}^n on a lower dimension subspace is a cluster we cannot conclude that the set K itself is a cluster. Another difficulty is that disjoint clusters in the n -dimensional space may have non-disjoint projections on lower-dimensional subspaces of \mathbb{R}^n , a phenomenon that we refer to as *occultation*.

Let C, D be two clusters in \mathbb{R}^n and let \mathbf{u} be a unit vector in the same space. To simplify the presentation assume that C and D are approximated by spheres of radius r_1 and r_2 , centered in the points \mathbf{c} and \mathbf{d} , respectively. The orthogonal projection of a set K on a vector \mathbf{u} is the set:

$$\text{proj}_{\mathbf{u}}(K) = \{\mathbf{u} \cdot \mathbf{x} | \mathbf{x} \in K\}.$$

An *\mathbf{u} -occultation* of the clusters C, D occurs if

$$\text{proj}_{\mathbf{u}}(C) \cap \text{proj}_{\mathbf{u}}(D) \neq \emptyset,$$

a situation which is represented in Figure 11

This is an inconvenient situation from our point of view since it fuses the two projections of C and D on the vector \mathbf{u} .

We need to evaluate the probability that an \mathbf{u} -occultation may occur for clusters since we will use cluster uni-dimensional projections for the identification of these clusters in \mathbb{R}^n . As before, we assume that \mathbf{u} is a unit random vector. The angle α between \mathbf{u} and the vector $\mathbf{d} - \mathbf{c}$ is uniformly distributed in the interval $[0, 2\pi]$. The discussion is essentially the same for projections on subspaces having an arbitrary dimensionality. Under the previous assumptions, an \mathbf{u} -occultation of the clusters C, D occurs when the length of the projection of the segment that joins \mathbf{c} to \mathbf{d} is inferior to $r_1 + r_2$; in other words if $|\mathbf{u} \cdot (\mathbf{d} - \mathbf{c})| = \|\mathbf{d} - \mathbf{c}\| |\cos \alpha| \leq r_1 + r_2$.

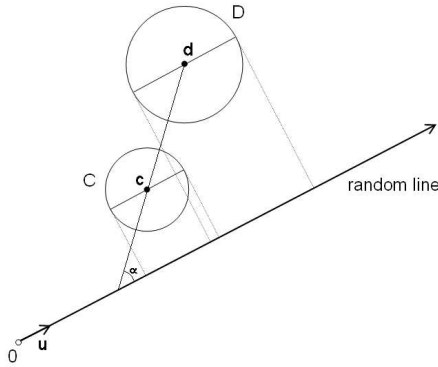


Fig. 1. Cluster Occultation

Consequently, the probability of an \mathbf{u} -occultation of the clusters is the number:

$$P\left(-\frac{r_1 + r_2}{\|\mathbf{d} - \mathbf{c}\|} \leq \cos \alpha \leq \frac{r_1 + r_2}{\|\mathbf{d} - \mathbf{c}\|}\right),$$

which is easily seen to equal to

$$1 - \frac{2}{\pi} \arccos\left(\frac{r_1 + r_2}{\|\mathbf{d} - \mathbf{c}\|}\right),$$

whenever $\|\mathbf{d} - \mathbf{c}\| \geq r_1 + r_2$, which happens when the clusters are sufficiently tight. Using the MacLaurin series expansion of $\arccos z$,

$$\arccos z = \frac{\pi}{2} - \left(z + \frac{z^3}{6} + \frac{3}{40} \frac{z^5}{5} + \dots\right),$$

we can approximate this value by $z = \frac{2(r_1+r_2)}{\pi\|\mathbf{d}-\mathbf{c}\|}$.

Let O_i be the event that takes place when an occultation occurs on the i -th projection of the random frame, for $1 \leq i \leq n$. We need to evaluate the probability that there is at least one projection that avoids the occultation, that is, $P(\overline{O_1} \cup \dots \cup \overline{O_n}) = 1 - P(O_1 \cap \dots \cap O_n)$. Assuming that the events O_1, \dots, O_n are independent we have

$$P(\overline{O_1} \cup \dots \cup \overline{O_n}) = 1 - \left(\frac{2(r_1 + r_2)}{\pi \|\mathbf{d} - \mathbf{c}\|}\right)^n.$$

Of course, the independence supposition does not hold in reality. We adopt it here to obtain an estimate that is plausible and is verified by experimental work.

Thus, the probability that there is a dimension that avoids occultation is increasing quite rapidly with the number of dimensions and with the inter-cluster separation. This shows the usefulness of the randomly chosen frame in separating, at least partially, and with a degree of uncertainty, clusters that may not be differentiated through their projections on the initial system of coordinates.

3 The Clustering Algorithm

Our algorithm has a heuristic nature. The input consists of a numerical n -dimensional data set D and entails two phases: in the first phase we apply random projections to the data set and we obtain the primary clusters; in the second phase we refine the clustering by using two processes: bimodulation and cluster expansion.

The projection phase begins with a randomly chosen orthogonal $n \times n$ -matrix \mathbf{H} of real numbers whose rows are the $\mathbf{u}_1, \dots, \mathbf{u}_n$. Then, the data set D is projected onto each of the n dimensions of the newly chosen random base, resulting in n histograms. We begin by clustering the points of each of the selected uni-dimensional projections.

Each i -th histogram contains a number of k bins of width ℓ_i and the choice of k depends on the size of D . For example, for $|D| = 10^4$ we used $k = 50$. On each histogram we identify the peaks and the valleys. The peaks $h_1^i, \dots, h_{m_i}^i$ of this histogram may correspond to n -dimensional clusters.

Suppose that the peak h_j^i of the i -th projection is located between the lows l_j^i and l_{j+1}^i . Then, the set C_j^i consists of the points that belong to the p bins located at the left of h_j^i whose heights vary between $\beta h_j^i + (1 - \beta)l_j^i$ and h_j^i and the q bins located at the right of h_j^i whose heights vary between $\beta h_j^i + (1 - \beta)l_{j+1}^i$ and h_j^i (see Figure 2). Here β is a parameter chosen by the user that allows us to guarantee a certain cluster density.

Note that the density of the cluster $\text{proj}_i(S) \cap C_j^i$ is at least

$$\frac{h_j^i + p[\beta h_j^i + (1 - \beta)l_j^i] + q[\beta h_j^i + (1 - \beta)l_{j+1}^i]}{(p + q + 1)\ell_i},$$

which is easily seen to be at least $\frac{h_j^i \beta}{\ell_i}$. So, if we choose

$$\beta \geq \max\left\{\frac{\delta_1 \ell_i}{\min h_j^i} \mid 1 \leq i \leq n\right\},$$

we guarantee that the uni-dimensional clusters have the minimal density δ_1 . The choice of δ_1 is determined, as we shall see by the parameter δ .

The quality of a projection is evaluated using the product between the average height of peaks and the logarithm of the number of peaks of the histogram. Only a percentage of the dimensions that correspond to these top histograms are retained for the next phase. Initially we seek to obtain a clustering that corresponds to these projections. In our experiments we used the top 10% of the histograms, a choice that is supported by our experiments (see Section 4.2).

Suppose that the peaks of the i -th random projection correspond to the intervals $C_1^i, \dots, C_{p_i}^i$. Let t be the number of top projections. We use a file \mathcal{F} that contains records having $1 + n + t$ components, whose structure is shown in Table 1. Each record represents one of the points $\mathbf{x} = (x_1, \dots, x_n)$ to be clustered and contains a point

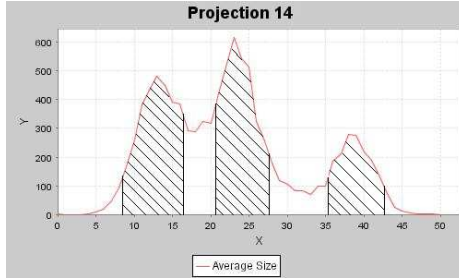


Fig. 2. Intervals around peaks in a projection

Table 1. The Structure of the file \mathcal{F}

\mathcal{F}						
Point id.	x_1	\cdots	x_n	$B(1, \mathbf{x})$	\cdots	$B(t, \mathbf{x})$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
h	a_1	\cdots	a_n	b_1	\cdots	b_t
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots

identifier, the original n coordinates x_1, \dots, x_n , and, for each projection i , a number $B(\mathbf{x}, i)$ defined by:

$$B(\mathbf{x}, i) = \begin{cases} j & \text{if the } i^{\text{th}} \text{ projection of} \\ & \mathbf{x} \text{ belongs to } C_j^i, \\ 0 & \text{otherwise.} \end{cases}$$

Records containing at least one 0 are discarded since they contain points that at this stage of the algorithm are not yet affiliated with any cluster. Then, the file \mathcal{F} is sorted on the fields $B(1, \mathbf{x}), \dots, B(t, \mathbf{x})$. Each set of points that correspond to a vector (b_1, \dots, b_t) corresponds to a set

$$C_{b_1 \dots b_t}^{i_1 \dots i_t} = C_{b_1}^{i_1} \times \dots \times C_{b_t}^{i_t}$$

which we regard as a constituent of the clustering. The condition

$$\frac{|\text{proj}_{i_1 \dots i_t}(S) \cap C_{b_1 \dots b_t}^{i_1 \dots i_t}|}{m(C_{b_1 \dots b_t}^{i_1 \dots i_t})} \geq \delta$$

insures that the clusters

$$\text{proj}_{i_1 \dots i_t}(S) \cap C_{b_1 \dots b_t}^{i_1 \dots i_t},$$

where $\text{proj}_{i_1 \dots i_t}(S)$ is the projection of S on the dimensions $i_1 \dots i_t$ of the random frame of coordinates will have the minimum density δ provided by the user. In our experiments we used $\delta = 0.01$ which reflects our decision of regarding clusters that contain less than 1% as consisting of outliers.

Note that

$$|\text{proj}_{i_1 \dots i_t}(S) \cap C_{b_1 \dots b_t}^{i_1 \dots i_t}| \leq \min_r |\text{proj}_{i_r}(S) \cap C_{b_r}^{i_r}|.$$

Therefore, the minima density condition imposed on the clusters implies that the uni-dimensional density δ_1 must be at least $\delta \ell^{t-1}$, where ℓ is the width of the bins defined above.

The time required to compute the histograms is $O(n^2N)$, where n is the number of dimensions and N is the number of points to be clustered. The cost of sorting the file \mathcal{F} is $O(N \log N)$, which brings the total cost of the algorithm to $O(n^2N + N \log N)$. Thus, the asymptotic cost of the algorithm is $O(N \log N)$; however, when the number of dimensions is important relative to the logarithm of the number of points the $O(n^2N)$ component is not negligible.

The post-processing of the clusters described below does not alter this asymptotic evaluation.

Assigning points left outside the clusters, to the extent that this is possible, is achieved using multiple random projections. Suppose that two random projections yield two clusterings:

$$\kappa = \{C_1, \dots, C_p\} \text{ and } \mu = \{D_1, \dots, D_q\}$$

and let $U_i = \text{UNC}(\mu) \cap C_i$ for $1 \leq i \leq p$ and $V_j = \text{UNC}(\kappa) \cap D_j$ for $1 \leq j \leq q$. Clusterings obtained by distinct random projections may be used to produce better clusterings by a process that will be referred here as *bimodulation*.

Let $\text{PART}(S)$ be the set of partitions of a set S and let $\text{CL}(S)$ be the set of clusterings of same set S . Every clustering $\kappa = \{C_1, \dots, C_p\}$, defines a partition $\pi_\kappa = \{C_1, \dots, C_p, \text{UNC}(\kappa)\}$ of the set S .

Let $d : \text{PART}(S) \times \text{PART}(S) \rightarrow \mathbb{R}$ be a distance defined on the set of partitions of the set S . A *d-bimodulation* is a mapping: $\Psi : \text{CL}(S) \times \text{CL}(S) \rightarrow \text{CL}(S) \times \text{CL}(S)$ such that if $\Psi(\kappa, \mu) = (\kappa', \mu')$, then $d(\kappa', \mu') \leq d(\kappa, \mu)$. In other words, an application of a bimodulation to a pair of clusterings results in a new pair of clusterings whose partitions are closer to each other than the partitions associated to the initial pair of clusterings. We can use as a distance between partitions the Barthélemy-Montjardet distance introduced in [4]. If π, σ are two partitions in $\text{PART}(S)$ given by:

$$\pi = \{K_1, \dots, K_m\} \text{ and } \sigma = \{H_1, \dots, H_n\},$$

then the distance between π and σ is given by:

$$d(\pi, \sigma) = \sum_{i=1}^m |K_i|^2 + \sum_{j=1}^n |H_j|^2 - 2 \sum_{i=1}^m \sum_{j=1}^n |K_i \cap H_j|^2.$$

It is possible to show that for any two clusterings $\kappa = \{C_1, \dots, C_p\}$ and $\mu = \{D_1, \dots, D_q\}$ a *d-bimodulation* can be defined by adding to each cluster C_i the set of objects located in the cluster D_j that has the largest intersection with C_i and applying a similar expansion to the clusters D_j .

The second method applied for cluster post-processing is using the minimum bounding hyper-rectangle $\text{MBH}(C)$ of a cluster C . Suppose that:

$$\text{MBH}(C) = [a_1, b_1] \times \dots \times [a_r, b_r].$$

The *density* of C is defined as the number:

$$\text{dens}(C) = \frac{|C|}{\text{vol}(\text{MBH}(C))}.$$

An ϵ -*expansion* of C is the set $C^\epsilon = C \cup L^\epsilon$, where

$$L^\epsilon = \text{UNC}(\kappa) \cap ([a_1 - |a_1|\epsilon, b_1 + |b_1|\epsilon] \times \dots \\ \dots \times [a_r - |a_r|\epsilon, b_r + |b_r|\epsilon]).$$

If $C_i^\epsilon \cap C_j^\epsilon \neq \emptyset$, then we assign the points of K_ϵ to the cluster that has the larger density among the clusters C_i^ϵ or to C_j^ϵ .

Experimental results show that these post-processing techniques improve significantly the quality of the clustering; this is clearly visualized by the improvement of the quality of the image segmentation that we discuss in Section 4.3.

4 Experimental Results

We performed experimental work on three types of data: synthetic data consisting of randomly-generated points in \mathbb{R}^n , synthetic images containing colored regions randomly distributed, and, finally, real images. The implementations of k -means [13] and DBSCAN [15] provided by the open-source WEKA package [18] were used for performance comparisons.

To evaluate the extent to which the algorithm retrieves the original clusters we computed several classification-oriented measure of cluster validity (see [16], p. 549). We assume that we start with r clusters K_1, \dots, K_r and the clusters retrieved by the algorithm are C_j , where $1 \leq j \leq q$. Also, the probability that an object of the cluster C_j belongs to K_ℓ is the number $p_{j\ell} = \frac{|C_j \cap K_\ell|}{|D_j|}$, which is also known as the *precision* of C_j relative to K_ℓ .

4.1 Experiments on Synthetic Data

We tested our technique on a data set containing 10000 points in \mathbb{R}^{30} distributed in four clusters: K_1, K_2, K_3, K_4 . We recaptured a major part of the data set, as shown in Table 2.

Table 2 represents the intersections between the original clusters K_1, \dots, K_4 (which correspond to the columns of the table) with the clusters C_1, \dots, C_4 obtained by our algorithm. The last row represents the points that the algorithm left outside the clusters. Before the postprocessing phase a substantial fraction of the points of the initial clusters are placed into clusters (almost 50%); however, many points are left unaffiliated with any of the clusters. These points are classified using the second phase of the algorithm.

After bimodulation and a 5% expansion the data distribution looks as shown in Table 3.

We further tested our algorithm using a similar data set (10000 points in \mathbb{R}^{30}) and added 10% of noisy data. The results are shown in Table 4, which shows that the noise has little effect on the clusters.

Table 2. Intersections between initial clusters and retrieved clusters before postprocessing

C_i identified	K_1	K_2	K_3	K_4
C_1	0	54	0	0
C_2	0	305	0	0
C_3	0	272	0	0
C_4	0	0	0	274
C_5	0	0	0	1170
C_6	2	0	74	0
C_7	1103	0	0	0
C_8	138	0	0	0
UNC(<i>data</i>)	2094	1083	712	2146

Table 3. Intersections between initial clusters and retrieved clusters after postprocessing

C_i identified	K_1	K_2	K_3	K_4
C_1	0	514	2	0
C_2	0	312	0	0
C_3	0	801	9	0
C_4	0	0	0	1189
C_5	0	0	0	2930
C_6	30	85	770	41
C_7	1761	0	0	0
C_8	1543	0	0	0
UNC(<i>data</i>)	3	2	5	5

Table 4. Intersections between initial clusters and retrieved clusters after introduction of noise

Clusters identified	K_1	K_2	K_3	K_4	Noise
C_1	0	2885	3	0	8
C_2	0	2	803	0	3
C_3	0	0	0	1929	10
C_4	1056	0	0	0	14
Data outside clusters	328	1472	591	931	965

We applied the k -means and the DBSCAN algorithms to the same data sets and obtained similar results for synthetic data without noise containing four clusters. Then, we tested these algorithms on a data set containing four clusters to which 10% noise was added. Several runs using the k -means algorithm with $k = 4$ result sometimes in having the noise distributed among each of the four clusters and, on occasions, producing 1 to 3 clusters with the remaining classes containing noise. The results of DBSCAN are similar to those of k -means; however, the misclassifications are much less frequent and the noise is well detected.

The application of the three algorithms to a database containing 11,000 objects in with 30 dimensions results in computation times of 750s, 1700 s, and 9s for our algorithm, the k -means algorithm and the DBSCAN algorithm, respectively. Our time is less than half of the DBSCAN. The k -means algorithm is much faster, but, as we shall see, has a rather bad precision and recall in experiments on synthetic images.

4.2 Experiments on Synthetic Images

In a second series of experiments we tested the algorithm on Mondrian-like images [14] containing randomly distributed and randomly colored rectangles. The typical images used in these experiments contained between 10 and 40 such regions and we show an example of an image in Figure 3.

The objectives of this series of experiments were to demonstrate that the algorithm can retrieve the original clusters and also, to test the behavior of the algorithm on data with a larger number of dimensions. Starting from an image containing $240 \times 320 = 76,800$ pixels represented as a set of points in \mathbb{R}^5 we grouped the pixels into 4×4

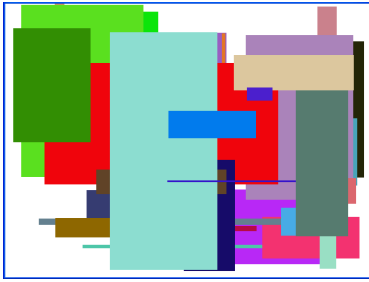


Fig. 3. Example of an image

True positive TP	Colored points retrieved as colored	63.29%
False positive FP	White points retrieved as colored	0.09%
True negative TN	White points retrieved as white	28.9%
False negative FN	Colored points retrieved as white	7.79%

Fig. 4. Terms used in algorithm evaluation

squares containing 16 pixels. Each square was represented as a vector in \mathbb{R}^{80} and we worked with sets of 9600 points in \mathbb{R}^{80} .

In a first phase we examined the capability of our algorithm to differentiate between the colored regions which we treat as clusters) and the white pixels. We are using the top five projections with an expansion factor of 5%. The terms used for this evaluation are shown in Figure 4. The precision and recall for this type of evaluations are given by

$$\text{Precision} = \frac{TP}{TP + FP} = 0.99 \text{ and } \text{Recall} = \frac{TP}{TP + FN} = 0.89,$$

respectively. The F_1 measure that is the harmonic average of precision and recall is 0.94. These numbers indicate a high capability of our algorithm in identifying points that belong to clusters.

The dependency of the precision, recall and F_1 measures are shown in Figures 5-7 respectively.

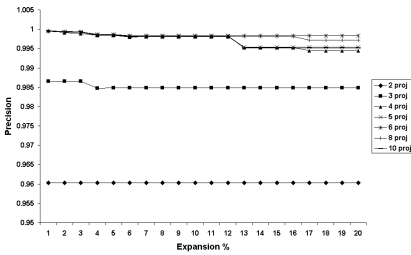


Fig. 5. Dependency of precision on ϵ

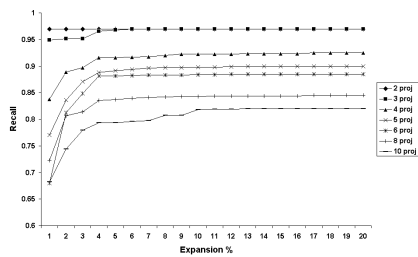


Fig. 6. Dependency of recall on ϵ

By contemplating these figures it becomes apparent that there is no substantial improvement of the recall or of the F_1 factor when expansion is greater than 6% and the number of projections considered is greater than 6 (see Figures 7 and 6). This observation informs the experiments described in the next section.

The time requirements of the algorithm were validated in experiments including data in \mathbb{R}^{20} and in \mathbb{R}^{80} (see Figure 8). The results shown represent averages over 4-fold

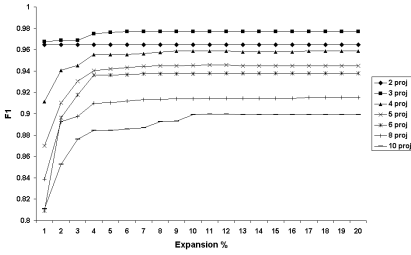


Fig. 7. Dependency of F_1 on ϵ

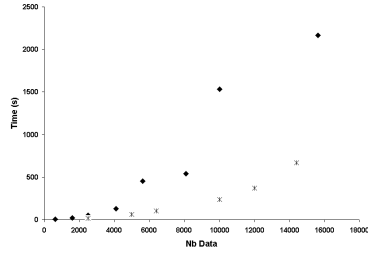


Fig. 8. Dependency of time (sec.) on the number of objects

complete applications of the algorithm including post-processing with different random projection frames. They are consistent with our previous asymptotic estimate of $O(N \log N)$.

In Table 5 we compare the precision, recall and F_1 measure for k -means, DBSCAN, and for our algorithm.

Table 5. Time measures

	Algorithm		
	k -means	DBSCAN	Our algorithm
TP (%)	62	58	63.3
FP (%)	28	3	0.1
TN (%)	10	34	28.9
FN (%)	1	5	7.8
Precision	0.69	0.95	0.99
Recall	0.98	0.92	0.89
F_1	0.81	0.94	0.94

For synthetic images the precision of the k -means algorithm is rather low even if k is chosen to obtain the best results. On the other hand, the results of DBSCAN and of our algorithm are comparable; we obtain a better precision but a lower recall which results in similar values for the F_1 measures.

The advantage of our algorithm over DBSCAN is a better time performance, which is more evident with the increase in the size of the data set.

4.3 Experiments on Real Images

To contemplate possible applications of our algorithm to multimedia data set we used the data set underlying the left picture shown in Figure 9. This data set contains 32,000 points in \mathbb{R}^5 . The dimensions correspond to the two spatial coordinates and the three color components of each pixel (red, green and blue). The two following images of

the same figure correspond to two clusterings obtained using our random projection algorithm.

We have applied the bimodulation technique to the clusters contained in the second and third images of Figure 9 which consist of 10 and 12 clusters, respectively; the images that correspond to the resulting clusterings are shown in the fourth and fifth images of Figure 9. The clusterings shown in these two images consist of 11 and 13 clusters, respectively. One can visually remark the improvement of certain features shown in these clusters. However, an important fraction of the data points still remain unclassified; these unclassified data correspond to the white spots of the illustrations.

Finally, we used the ϵ -expansion of the minimally bounding rectangles of the clusters. The new clusters obtained by applying a 10% expansion are presented in the last two images shown in Figure 9.

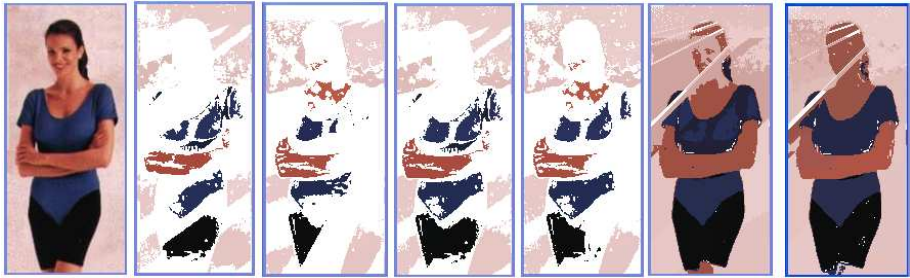


Fig. 9. Images obtained at different phases of our algorithm

The quality of the last two images is clearly improved over the others images. This observation suggests that our clustering techniques have the potential of being helpful in image segmentation .

5 Conclusions

We proposed a new method of clustering using random projections. The algorithm consists in two phases: a projection phase (which creates uni-dimensional histograms and aggregates these histograms to produce the initial clusters) and a post-processing phase that improves the clusterings using two supplementary techniques: bimodulation and ϵ -expansion. The time requirement of the algorithm is $O(N \log N)$, where N is the number of objects subjected to clustering. The algorithm has a potential for being useful for multimedia applications, which will be the focus of our future investigations.

We will investigate future directions of cluster post-processing as well as more refined ways of combining projection histograms. Finding optimal values for the parameters chosen in the execution of the algorithm based on the statistical distribution of the set of objects remains an open problem.

References

1. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of the ACM-SIGMOD Int. Conf. Management of Data, pp. 94–105. ACM Press, New York (1998)
2. Agarwal, P., Mustafa, N.H.: k-means projective clustering. In: Proceedings of PODS, pp. 155–165 (2004)
3. Aggarwal, C.C., Procopiuc, C., Wolf, J.L., Yu, P.S., Park, J.S.: Fast algorithms for projected clustering. In: Proceedings of ACM-SIGMOD Conference on Management of Data, pp. 61–72. ACM Press, New York (1999)
4. Barthélemy, J.P., Leclerc, B.: The median procedure for partitions. In: Partitioning Data Sets. American Mathematical Society, pp. 3–14. Providence, RI (1995)
5. Chaudhri, A.B., Unland, R., Djeraba, C., Lindner, W. (eds.): EDBT 2002. LNCS, vol. 2490. Springer, Heidelberg (2002)
6. Dasgupta, S., Gupta, A.: An elementary proof of the johnson-lindenstrauss lemma. Technical Report TR-99-006, International Computer Science Institute (1999)
7. Djeraba, C. (ed.): Multimedia Mining - A Highway to Intelligent Multimedia Documents. Kluwer, Dordrecht (2003)
8. Frankl, P., Maehara, H.: The johnson-lindenstrauss lemma and the sphericity of some graphs. *J. Comb. Theory B* 44, 355–362 (1988)
9. Jain, A.K., Dubes, R.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs (1988)
10. Jain, A.K., Flynn, P.J.: Image segmentation using clustering. In: Advances in Image Understanding: A Festschrift for Aziel Rosenfeld, Piscataway, NJ, pp. 65–83. IEEE Press, Los Alamitos (1996)
11. Johnson, W.B., Lindenstrauss, J.: Extensions of lipshitz mappings into hilbert spaces. *Contemporary Mathematics* 26, 189–206 (1984)
12. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. *ACM Computing Surveys* 31, 264–323 (1999)
13. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, pp. 281–297. University of California Press, California (1967)
14. Mondrian, P.: <http://artchive.com/artchive/M/mondrian.html>
15. Sander, J., Ester, M., Kriegel, H.P., Xu, X.: Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery, an International Journal* 2, 169–194 (1998)
16. Tan, P.N, Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson/Addison-Wesley, Boston (2006)
17. Vempala, S.S.: The Random Projection Method. American Mathematical Society. Providence, Rhode Island (2004)
18. Witten, I.H., Frank, E.: Data Mining - Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
19. Zaïane, O.R., Simoff, S.J., Djeraba, C. (eds.): MDM/KDD 2002 and KDMCD 2002. LNCS (LNAI), vol. 2797. Springer, Heidelberg (2002)

Lightweight Clustering Technique for Distributed Data Mining Applications*

Lamine M. Aouad, Nhien-An Le-Khac, and Tahar M. Kechadi

School of Computer Science and Informatics
University College Dublin - Ireland
{lamine.aouad,an.le-khac,tahar.kechadi}@ucd.ie

Abstract. Many parallel and distributed clustering algorithms have already been proposed. Most of them are based on the aggregation of local models according to some collected local statistics. In this paper, we propose a lightweight distributed clustering algorithm based on minimum variance increases criterion which requires a very limited communication overhead. We also introduce the notion of distributed perturbation to improve the globally generated clustering. We show that this algorithm improves the quality of the overall clustering and manage to find the real structure and number of clusters of the global dataset.

1 Introduction

Clustering is one of the fundamental technique in data mining. It groups data objects based on information found in the data that describes the objects and their relationships. The goal is to optimize similarity within a cluster and the dissimilarities between clusters in order to identify interesting structures in the underlying data. This is a difficult task in unsupervised knowledge discovery and there is already a large amount of literature in the field ranging from models, algorithms, validity and performances studies, etc. However, there is still several open questions in the clustering process including the optimal number of clusters, how to assess the validity of a given clustering, how to allow different shapes and sizes rather than forcing them into balls and shapes related to the distance functions, how to prevent the algorithms initialization and the order in which the features vectors are read in from affecting the clustering output, and how to find which clustering structure in a given dataset, i.e why would a user choose an algorithm instead of another. Most of these issues come from the fact that there is no general definition of what is a cluster. In fact, algorithms have been developed to find several kinds of clusters; spherical, linear, dense, drawnout, etc.

In distributed environments, clustering algorithms have to deal with the problem of distributed data, computing nodes and domains, plural ownership and users, and scalability. Actually, moving the entire data to a single location for

* This study is part of ADMIRE [15], a distributed data mining framework designed and developed at University College Dublin, Ireland.

performing a global clustering is not always possible due to different reasons related to policies or technical choices. In addition, the communication efficiency of an algorithm is often more important than the accuracy of its results. In fact, communication issues are the key factors in the implementation of any distributed algorithm. It is obvious that a suitable algorithm for high speed network can be of little use in WAN-based platforms. Generally, it is considered that an efficient distributed algorithm needs to exchange a few data and avoids synchronization as much as possible.

In this paper, we propose a lightweight distributed clustering technique based on a merging of independent local subclusters according to an increasing variance constraint. This improves the overall clustering quality and finds the number of clusters and the global inherent clustering structure in the whole dataset. However, a proper maximum increasing value has to be selected. This can be deduced from the problem domain or found out using various methods. The rest of the paper is organized as follows, the next section surveys some previous parallelization and distribution efforts in the clustering area. Then, section 3 presents our distributed algorithm. Section 4 shows some experimental results and evaluations, and highlights directions for future work and versions. Finally, section 5 concludes the paper.

2 Related Work

This section survey some works in parallel and distributed clustering, and discusses the latest projects and proposals especially regarding grid-based approaches.

Clustering algorithms can be divided into two main categories, namely partitioning and hierarchical. Different elaborated taxonomies of existing clustering algorithms are given in the literature. Details about these algorithms is out of the purpose of this paper, we refer the reader to [8] and [19]. Many parallel clustering versions based on these algorithms have been proposed [3][4][5][6][13][20], etc. In [3] and [13], message-passing versions of the widely used k-means algorithm were proposed. In [4] and [20], the authors dealt with the parallelization of the DBSCAN density based clustering algorithm. In [5] a parallel message passing version of the BIRCH algorithm was presented. In [6], the authors introduced a parallel version of a hierarchical clustering algorithm, called MPC for Message Passing Clustering, which is especially dedicated to Microarray data. Most of the parallel approaches need either multiple synchronization constraints between processes or a global view of the dataset, or both.

The distributed approaches are different, even many of the proposed distributed algorithms are based on algorithms which were developed for parallel systems. Actually, most of them typically act by producing local models followed by the generation of a global model by aggregating the local results. The processes participating to the computation are independent and usually have the same computation level. After this phase, the global clustering is obtained based on only local models, without a global view of the whole dataset. All these

algorithms are then based on the global reduction of so-called sufficient statistics, probably followed by a broadcast of the results. Some works are presented in [9] [10] [11] [12] [21], mostly related to the k-means algorithm or variants and the DBSCAN density based algorithm.

On the other hand, grid and peer-to-peer systems have emerged as an important area in distributed and parallel computing¹. In the data mining domain, where massive datasets are collected and need to be stored and performed, the grid can be seen as a new computational and large-scale support, and even as a high performance support in some cases. Some grid or peer-to-peer based projects and frameworks already exist or are being proposed in this area; Knowledge Grid [2], Discovery Net [7], Grid Miner [14], ADMIRE [15], etc. Beyond the architecture design of these systems, the data analysis, integration or placement approaches, the underlying middleware and tools, etc. the grid-based approach needs efficient and well-adapted algorithms. This is the motivation of this work.

3 Algorithm Description

This section describes the distributed algorithm and gives some formal definitions. The key idea of this algorithm is to choose a relatively high number of clusters locally (which will be called subclusters in the rest of the paper), or an optimal local number using an approximation technique, and to merge them at the global level according to an increasing variance criterion which require a very limited communication overhead. All local clustering are independent from each other and the global aggregation can be done independently, from and at any initial local process.

3.1 Algorithm Foundations

At the local level, the clustering can be done by different clustering algorithms depending on the characteristics of the data. This includes k-means, k-harmonic-means, k-medoids, or their variants, or using the statistical interpretation with the expectation-maximization algorithm which finds clusters by determining a mixture of Gaussian distributions. The merging process of the local subclusters at the global level exploits locality in the feature space, i.e. the most promising candidates to form a global cluster are subclusters that are the closest in the feature space, including subclusters from the same site. Each participating process can perform the merging and subtract the global clusters formation, i.e. which subclusters are subject to form together a global cluster.

Before describing the algorithm itself, we first give developments on some used notions. A global cluster border represents local subclusters at its border. These are susceptible to be isolated and added to another global cluster in order to

¹ The designation 'parallel' is used here to highlight the fact that the computing tasks are interdependent, which is not necessarily the case in distributed computing.

contribute to an improvement of the clustering output. These subclusters are referred to as perturbation candidates. Actually, the initial merging order may affect the clustering output, as well as the presence of non well-separated global clusters, this action is intended to reduce the input order impact. The global clusters are then updated. The border is collected by computing the common Euclidean distance measure. The b farthest subclusters are then the perturbation candidates, where b is a user predefined number which depends on the chosen local number of clusters. Furthermore, multi-attributed subclusters are naturally concerned by this process.

The aggregation part of the algorithm starts with $\sum_{i \in s} k_i$ subclusters, where s is the number of sites involved and k_i , for $i = 1, \dots, s$, are the local numbers of clusters in each site. Each process has the possibility to generate a global merging. An important point here is that the merging is logical, i.e. each local process can generate correspondences, i.e. labeling, between local subclusters, without necessarily constructing the overall clustering output. That is because the only bookkeeping needed from the other sites are centers, sizes and variances. The aggregation is then defined as a labeling process between local subclusters in each participating site. On the other hand, the perturbation process is activated if the merging action is no longer applied. b candidates are collected for each global cluster from its border, which is proportional to the overall size composition as quoted before. Then, this process moves these candidates by trying the closest ones and with respect to the gain in the variance criterion when moving them from the neighboring global clusters. In the next section we will formally define the problem, notions and criterions.

3.2 Definitions and Notations

This section formalizes the clustering problem and the notions described in the previous section. Let $X = \{x_1, x_2, \dots, x_N\}$ be a dataset of N elements in the p -dimensional metric space. The problem is to find a clustering of X in a set of clusters, denoted by $C = \{C_1, C_2, \dots, C_M\}$. The most used criterion to quantify the homogeneity inside a cluster is the variance criterion, or sum-of-squared-error criterion:

$$S = \sum_{i=1}^M E(C_i)$$

where

$$E(C) = \sum_{x \in C} \|x - u(C)\|^2$$

and

$$u(C) = \frac{1}{|C|} \sum_{x \in C} x$$

is the cluster mean.

Traditional constraint used to minimize the given criterion is to fix the number of clusters M to an a priori known number, as in the widely used

k-means, k-harmonicmeans, k-medoids or its variants like CLARA, CLARANS, etc. [16] [19] [22]. This constraint is very restrictive since this number is most likely not known in most cases. However, many approximation techniques exist such as the gap statistic which compares the change within cluster dispersion to that expected under an appropriate reference null distribution [17], or the index due to Calinski & Harabasz [1], etc. This can be used locally as quoted before. The imposed constraint here states that the increasing variance of the merging, or union, of two subclusters is below a dynamic limit $\sigma_{i,j}^{max}$. This parameter is defined to be twice the highest individual variance from subclusters C_i and C_j [18].

The border B_i of the global cluster C_i is the set of the b farthest subclusters from the generated global cluster center. Let $SC_i = \{scc_1, scc_2, \dots, scc_{n_i}\}$ be the set of the n_i subclusters centers merged into C_i . B_i is defined as:

$$B_i(b) = F(u(C_i), b, C_i, SC_i)$$

where

$$F(u(C_i), b, C_i, SC_i) =$$

$$\begin{cases} fsc(u(C_i), b, C_i, SC_i) \cup F(u(C_i), b - 1, C_i, SC_i - fsc(u(C_i), b, C_i, SC_i)), & b > 0 \\ \emptyset, & b = 0 \end{cases}$$

$fsc(u(C_i), b, C_i, SC_i)$ are the b farthest subclusters centers from $u(C_i)$:

$$fsc(u(C_i), b, C_i, SC_i) = \arg \max_{x \in SC_i} Euclidean(x, u(C_i))$$

These sets are then performed once the merging is no longer applied, and as quoted before, the multi-attributed subclusters will belong to it.

3.3 Summarized Algorithm

According to the previous definitions and formalism, the Algorithm 1 summarize the proposed approach. In the first step, local clustering are performed on each local dataset, the local number of clusters can be different in each site. Then, each local clustering in a site i gives as output k_i subclusters identified by a unique identifier, $cluster_{\{i,number\}}$ for $number = 0, \dots, k_i - 1$, and their sizes, centers and variances. At the end of local processes, local statistics are sent (5 - 9) to the chosen merging process j at step (4). Then, the subclusters aggregation is done in two phases; merging (10 - 12) and perturbation (13 - 24). In the latter phase, the border $B_i(b)$ is found (14 - 15), with $i \in k_g$, and b is a user defined parameter. For each $x \in B_i(b)$, the closet global cluster j is found and the new variance is computed. The actual perturbation, which still a labeling at the global level, is done if the new global variance is smaller (16 - 23). At the step (11), the new global statistics, namely the size, center and variance, are:

$$\begin{aligned} N_{new} &= N_i + N_j \\ c_{new} &= \frac{N_i}{N_{new}} c_i + \frac{N_j}{N_{new}} c_j \end{aligned}$$

Algorithm 1. Variance-based distributed clustering**Input:** X_i ($i = 1, \dots, s$) datasets, and k_i the number of subclusters in each site S_i **Output:** k_g global clusters, i.e the global subclusters distribution labeling

```

1: for  $i = 1$  to  $s$  do
2:    $LS_i = cluster(X_i, k_i)$ 
3: end for
4:  $j = select\_aggr\_site()$ 
5: for  $i = 1$  to  $s$  do
6:   if  $i \neq j$  then
7:      $send(sizes_i, centers_i, variances_i, j)$ 
8:   end if
9: end for

  at site the aggregation site  $j$ :
10: while  $var(C_i, C_j) < \sigma_{i,j}^{max}$  do
11:    $merge(C_i, C_j)$ 
12: end while
13: for  $i = 1$  to  $k_g$  do
14:    $find\_border(b, i)$ 
15:    $add\_multi\_attributed(i)$ 
16:   for  $x = 1$  to  $b$  do
17:      $j = closer\_global(x)$ 
18:      $var_{new} = var(C_i - C_x, C_j + C_x)$ 
19:     if  $var_{new} < var$  then
20:        $label(x, j)$ 
21:        $var = var_{new}$ 
22:     end if
23:   end for
24: end for

```

$$var_{new} = var_i + var_j + inc(i, j), \quad \forall C_i, C_j, i \neq j$$

where

$$inc(i, j) = \frac{N_i \times N_j}{N_i + N_j} \times Euclidean(C_i, C_j)$$

represents the increasing in the variance while merging C_i and C_j .

As in all clustering algorithms, the expected large variability in clusters shapes and densities is an issue. However, as we will show in the experiments section, the algorithm is efficient to detect well separated clusters and distribution with their effective distribution number. Otherwise, a clear definition of a cluster does not exist anymore. This is also an efficient way to improve the output for the k-means clustering and derivatives for example, without an a priori knowledge about the data or an estimation process for the number of clusters.

3.4 Performance Analysis

The computational complexity of this distributed algorithm depends on the algorithm used locally, the communication time, which is a gather operation and the merging computing time:

$$T = T_{comp} + T_{comm} + T_{merge}$$

If the local clustering is a k-means, the complexity T_{comp} is of order $O(N_i k_i d)$, where d is the dimension of the dataset. The communication time is the reduction of $3d \sum_{i \in s} k_i$ elements. Actually, the aggregation process gathers local information in order to perform the merging. If t_{comm}^i is the communication cost for one element from site i to the aggregation process j then

$$T_{comm} = 3d \sum_{i \in s, i \neq j} t_{comm}^i k_i$$

Since k_i is much less large than N_i , the generated communication overhead is very small.

The merging process is executed a number of times, say u . This is the number of iterations until the condition $var(C_i, C_j) < \sigma_{i,j}^{max}$ is no longer applied. This cost is then equal to $u \times t_{newStatistics}$, which corresponds to $O(d)$. This is followed by a perturbation process, which the cost is of order $O(bk_g k_i)$. Actually, since this process computes for each of the b chosen subcluster at the border of C_i , k_i distances for each of the k_g global clusters. The total cost is then:

$$T = O(N_i k_i d) + O(d) + O(bk_g k_i) + T_{comm}, \quad T_{comm} \ll O(N_i k_i d)$$

4 Experiments

In this section, we show the effectiveness of the proposed algorithm with some artificial and real datasets. We give a description of the data, the experimentation details and a discussion. As quoted before, the constraint parameter, i.e the maximum merging variance, is set up as twice the highest individual subcluster variance.

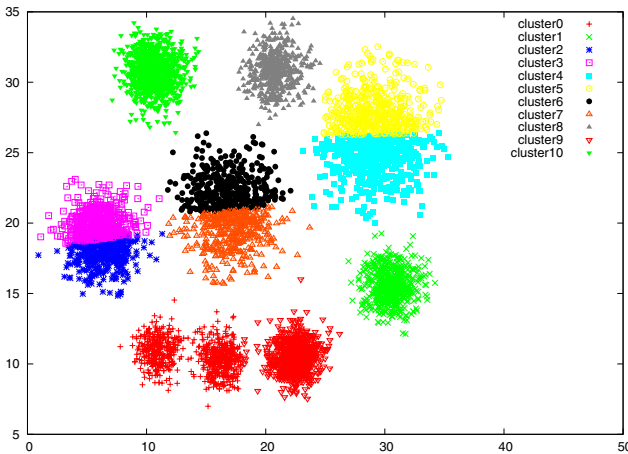


Fig. 1. Global k-harmonicmeans clustering using the gap statistic to find the optimal k of the dataset, $k = 11$

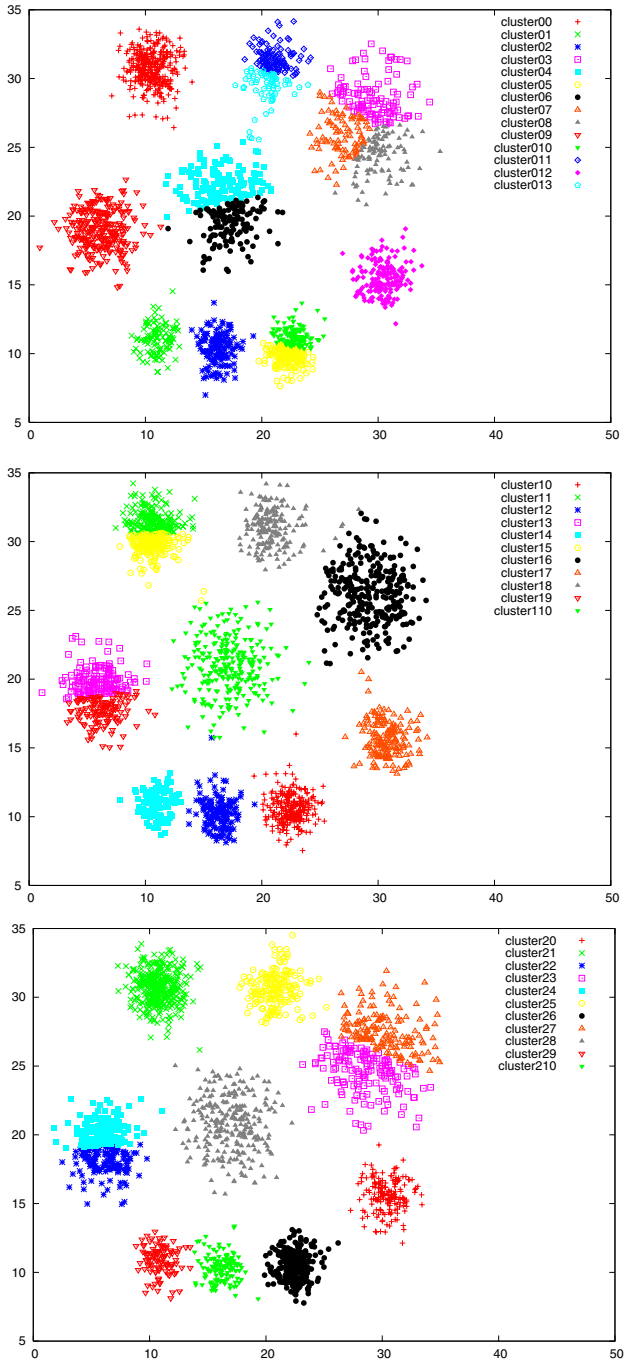


Fig. 2. Local k-harmonicmeans clustering in each process using the gap statistic to find the optimal number of clusters, $k_1 = 14$, $k_2 = 11$, and $k_3 = 11$

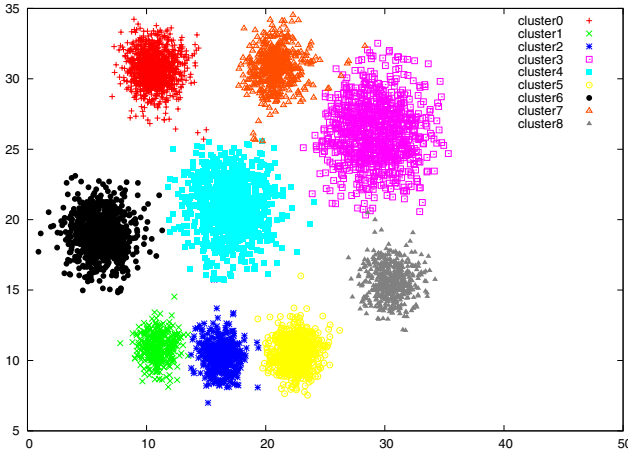


Fig. 3. Generated distributed clustering

4.1 Data Description

The first dataset is a generated random Gaussian distributions with 6000 samples. Figure 1 displays this dataset with an initial clustering using k-harmonicmeans and the gap statistic. The data was distributed in three sets as shown in Figure 2.

The second set is the well-known Iris dataset. It consists in three classes of irises (Iris setosa, Iris versicolor and Iris virginica) each characterized by 8 attributes and there is 150 instances. The set was randomly distributed as shown in Figure 4 (presented by the attributes “sepal area” and “petal area”). This figure shows also the initial local clustering using k-harmonicmeans with $k = 5$.

The last dataset is a census data available from the UC Irvine KDD Archive. It is derived from the one percent sample of the PUMS (Public Use Microdata Samples) person records from the full 1990 census sample. Our tests use a discretized version of this set. There are 68 attributes². The set originally contains 2458285 records reduced to 1998492 after elimination of doubled records. The data is distributed over 7 processes.

4.2 Evaluations and Discussion

The merging output of the first dataset is shown in Figure 3. This result finds the right number of clusters and their distribution independently of the local used clustering algorithm and the number of subclusters. The local number of clusters found using the gap statistic is 14 for the first set and 11 for the two other sets (cf. Figure 2). The gap statistic based implementation of the expectation-maximization algorithm give the same clustering output.

² The caseid is ignored during analysis. The list of attributes and the coding for the values can be found at <http://kdd.ics.uci.edu/>

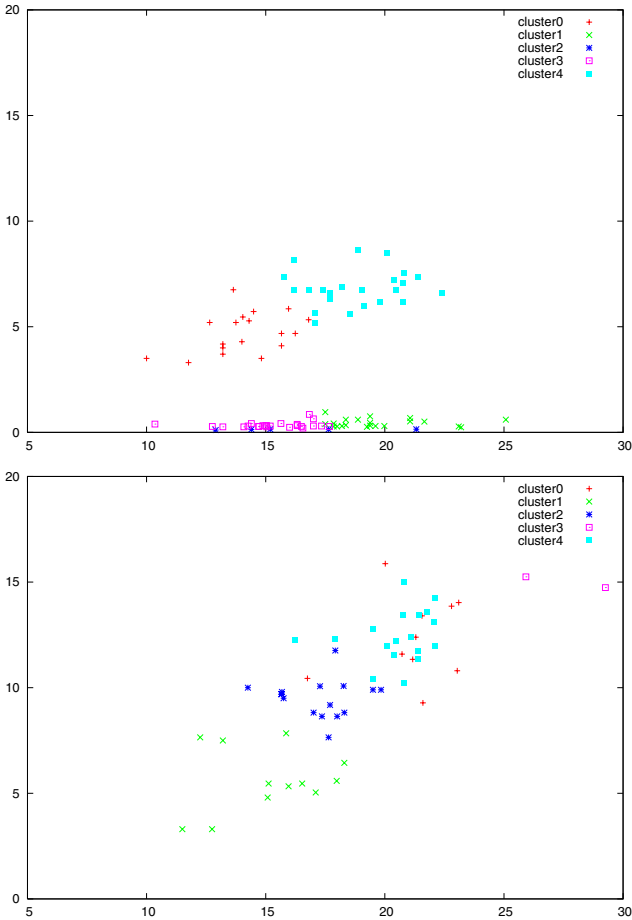
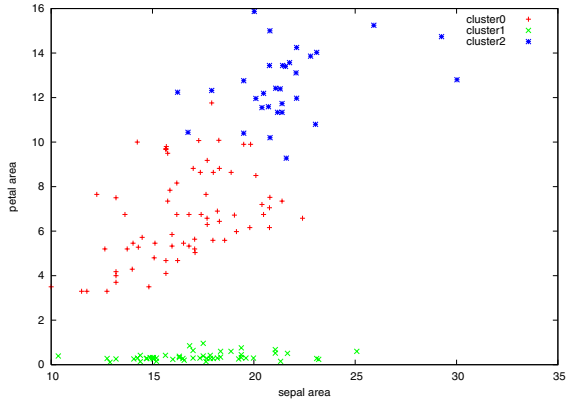


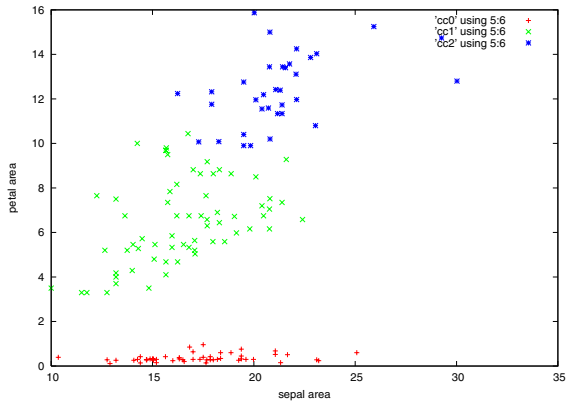
Fig. 4. Iris sub-sets and local clustering using k-harmonicmeans, $k_i = 5$, $i = 0, 1$

The resulting global clustering for the Iris dataset, and a global k-harmonicmeans clustering using the entire dataset, are given in Figure 5. The algorithm manages to find the class distribution of the Iris dataset, leading to 3 classes based on 5 or 7 local subclusters. However, because the k-harmonicmeans does not impose a variance constraint it could find a lower sum-of-squared-error which is the case here. These two examples show the independence from the nature and size of the initial clustering. Actually, if there is a real structure in the dataset then true clusters are found and joined together.

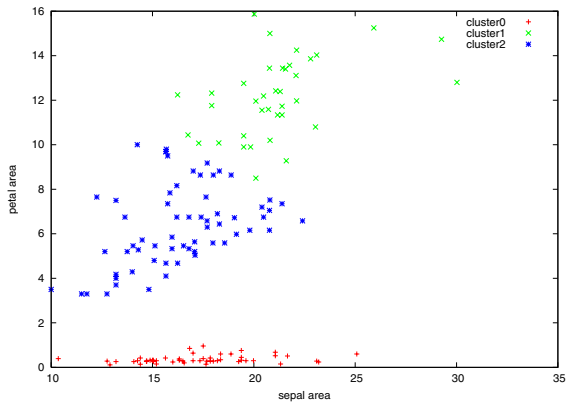
For the census dataset, the algorithm leads to 3 clusters based on 20 sub-clusters locally on 7 processes, and using all the attributes. The local clustering uses the k-means algorithm. This version is based on multiple k-means (user defined parameter) and keep the best output. Firstly, the Figure 6 shows the rank of the values of 9 attributes among the 68 for the whole dataset. The values



(a)



(b)



(c)

Fig. 5. The output using 5 (a) and 7 (b) subclusters, and a global clustering using k-harmonicmeans in (c)

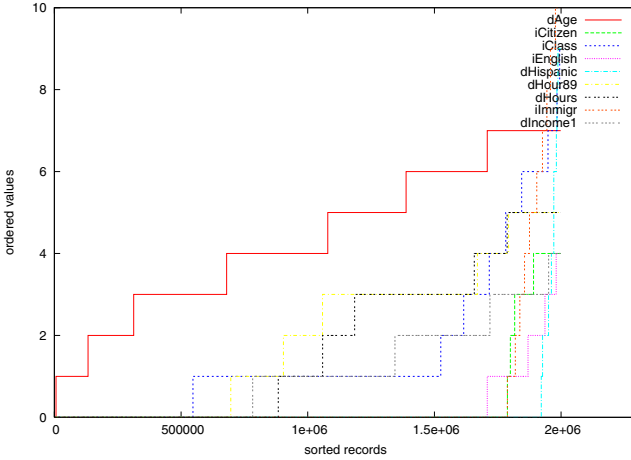


Fig. 6. Rank of the values of 9 attributes of the census database

distribution of two generated global clusters is given in Figure 7. Note that a global sequential clustering is not possible due to the memory restriction related to the fact that the whole data must fit in main memory. Most of the widely used clustering algorithms are concerned with this scalability issue. Beyond the signification of such clustering especially for this dataset and using the entire set of the categorical variables, this experiment shows the scalability of the proposed algorithm. Indeed, specific measurements on the dataset will take into account a specific set of variables. However, the Figure 7 shows, by the selected sorted attributes, different characteristics of these two global clusters concerning age or income for example. Still, the visualization mode does not allow to show the real measurement related to these attributes since they are sorted, which means that there is no true initial observations thereon.

In contrast to many other distributed algorithms, the presented one uses a simple global constraint, a very limited communication overhead, and does not need to know the data structure a priori. This algorithm is effective in finding proper clustering. However, future versions will take into account some other facts as considering the perturbation process during the merging operations and inside subclusters, or of whether or not multi-attributed clusters are present to consider a different approach at this level. Also, varying the constraint criterion could be considered as well as adding other similarity functions.

In fact, the merging process could perform a distance between the distributed subclusters. Each one could be described by additional objects and metrics, as the covariance matrices for example. A general definition of such a distance measure can be $d(x_i, x_j) = (c_j - c_i)^T A (c_j - c_i)$, where the inclusion of A results in weighting according to statistical properties of the features. Other distances or similarity measures include; Euclidean, Manhattan, Canberra, Cosine, etc. The general form of some of these distances is $d_{i,j} = [\sum_K |x_{ki} - x_{kj}|^N]^{\frac{1}{N}}$, and depending on N , the enclosed region takes different shapes. That is to say

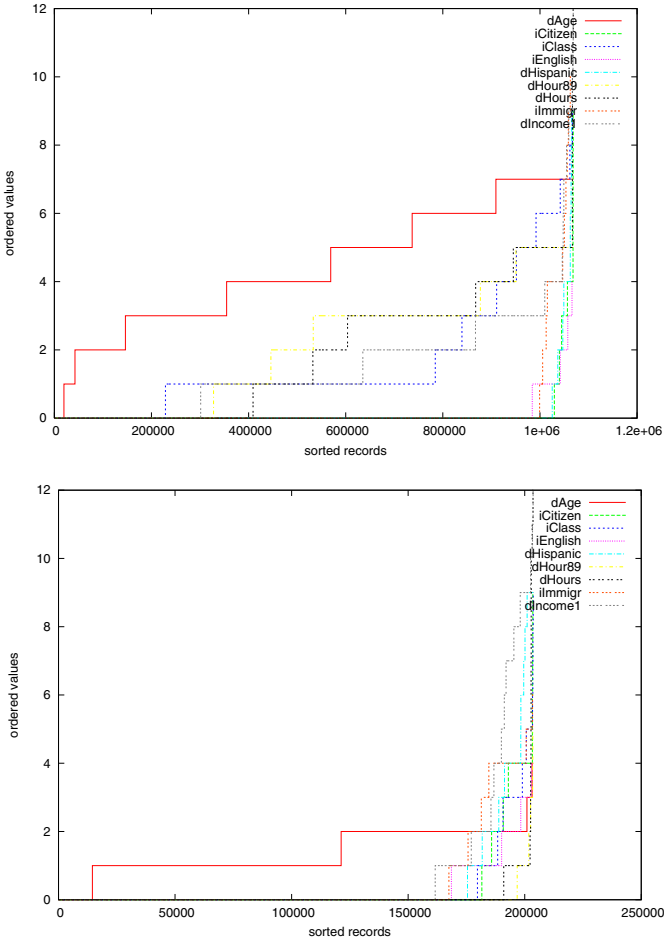


Fig. 7. Values distribution of two generated global clusters

that the merging process could take into account one or different proximity (i.e similarity or dissimilarity) function to improve the quality of the resulting clustering. This will be considered in future versions. However, the key issue is the selection of an appropriate function, especially, *which kind of measures for which kind of data?* Actually, general observations recommend some distances for some type of data, Euclidean-based for example for dense, continuous data. Still, no true rules exist and the user needs to be familiar and expertise his data.

5 Conclusion

In this paper, we evoked the need of efficient distributed and grid-based clustering algorithms. Actually, a huge effort has been made in sequential clustering

but there is only few algorithms which tackle the distribution problem especially in loosely coupled environments such as the grid. We proposed a lightweight distributed algorithm based on an increasing variance constraint. It clusters the data locally and independently from each other and only limited statistics about the local clustering are transmitted to the aggregation process which carries out the global clustering, defined as labeling between subclusters. This is done by means of a merging and a perturbation processes. The global model can then be broadcasted to all participating processes if needed, which will use it to label their subclusters.

The algorithm gives good performances at identifying well separated clusters and the real structure of the dataset. In fact, when data are not well separated, the notion of cluster is very confused and does not even exist in the literature. The number of clusters is also automatically found, this resolves the problem of estimating the number of clusters a priori. Furthermore, in addition to classical constraints in distributed clustering, related to the usually infeasible data centralization due to technical, security reasons or local policies, this algorithm can also tackle large and high dimensional datasets that cannot fit in memory since most of the clustering algorithms in literature require the whole data in the main memory and also tend to scale poorly as the size and dimension grow. Nevertheless, open issues could be considered as in the merging process or the choice of the possible better local models and algorithms, in addition to those described in the previous section.

References

1. Calinski, R.B., Harabasz, J.: A dendrite method for cluster analysis. *Communication in statistics* 3 (1974)
2. Cannataro, M., Congiusta, A., Pugliese, A., Talia, D., Trunfio, P.: Distributed Data Mining on Grids: Services, Tools, and Applications. *IEEE Transaction on System, Man, and Cybernetics* 34(6) (2004)
3. Dhillon, I.S., Modha, D.: A Data-Clustering Algorithm on Distributed Memory Multiprocessors. In: *Large-Scale Parallel Data Mining, Workshop on Large-Scale Parallel KDD Systems. SIGKDD* (1999)
4. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: *2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD)* (1996)
5. Garg, A., Mangla, A., Bhatnagar, V., Gupta, N.: PBIRCH: A Scalable Parallel Clustering algorithm for Incremental Data. In: *IDEAS'06. 10th International Database Engineering and Applications Symposium* (2006)
6. Geng, H., Deng, X., Ali, H.: A New Clustering Algorithm Using Message Passing and its Applications in Analyzing Microarray Data. In: *ICMLA'05. Proceedings of the Fourth International Conference on Machine Learning and Applications*, pp. 145–150. *IEEE Computer Society Press, Los Alamitos* (2005)
7. Ghanem, V.M., Kohler, Y.M., Sayed, A.J., Wendel, P.: Discovery Net: Towards a Grid of Knowledge Discovery. In: *Eight Int. Conf. on Knowledge Discovery and Data Mining* (2002)
8. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. *ACM Computing Surveys* (1999)

9. Januzaj, E., Kriegel, H-P., Pfeifle, M.: Towards Effective and Efficient Distributed Clustering. In: Int. Workshop on Clustering Large Data Sets. 3rd Int. Conf. on Data Mining, ICDM (2003)
10. Januzaj, E., Kriegel, H-P., Pfeifle, M.: DBDC: Density-Based Distributed Clustering. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., Ferrari, E. (eds.) EDBT 2004. LNCS, vol. 2992, Springer, Heidelberg (2004)
11. Januzaj, E., Kriegel, H-P., Pfeifle, M.: Scalable Density-Based Distributed Clustering. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) PKDD 2004. LNCS (LNAI), vol. 3202, Springer, Heidelberg (2004)
12. Jin, R., Goswami, A., Agrawal, G.: Fast and Exact Out-of-Core and Distributed K-Means Clustering. Knowledge and Information Systems 10 (2006)
13. Joshi, M.N.: Parallel K-Means Algorithm on Distributed Memory Multiprocessors. Technical report, University of Minnesota (2003)
14. Kicking, G., Hofer, J., Brezany, P., Tjoa, A.M.: Grid Knowledge Discovery Processes and an Architecture for their Composition. Parallel and Distributed Computing and Networks (2004)
15. Le-Khac, N.-A., Kechadi, M.T., Carthy, J.: ADMIRE framework: Distributed Data Mining on Data Grid platforms. In: ICSOFT 2006. first Int. Conf. on Software and Data Technologies (2006)
16. Ng, R.T., Han, J.: Efficient and Effective Clustering Methods for Spatial Data Mining. In: VLDB 1994. Proceedings of 20th International Conference on Very Large Data Bases, Santiago de Chile (1994)
17. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a dataset via the Gap statistic. Technical report, Stanford University (March 2000)
18. Veenman, C.J., Reinders, M.J., Backer, E.: A Maximum Variance Cluster Algorithm. IEEE Transactions on pattern analysis and machine intelligence 24(9) (2002)
19. Xu, R., Wunsch, D.: Survey of Clustering Algorithms. IEEE Transactions on Neural Networks 16 (2005)
20. Xu, X., Jager, J., Kriegel, H.-P.: A Fast Parallel Clustering Algorithm for Large Spatial Databases. Journal of Data Mining and Knowledge Discovery 3 (1999)
21. Zhang, B., Forman, G.: Distributed Data Clustering Can be Efficient and Exact. Technical report, HP Labs (2000)
22. Zhang, B., Hsu, M., Dayal, U.: K-Harmonic Means - A Data Clustering Algorithm. Technical report, HP Labs (1999)

Predicting Page Occurrence in a Click-Stream Data: Statistical and Rule-Based Approach

Petr Berka and Martin Labský

Department of Information and Knowledge Engineering,
University of Economics, W. Churchill Sq. 4, 130 67 Prague, Czech Republic
{berka,labsky}@vse.cz

Abstract. We present an analysis of the click-stream data with the aim to predict the next page that will be visited by an user based on a history of visited pages. We present one statistical method (based on Markov models) and two rule induction methods (first based on well known set covering approach, the other base on our compositional algorithm KEX). We compare the achieved results and discuss interesting patterns that appear in the data.

1 Introduction

Our work described in the paper fits into the area of web usage mining. Web usage mining mines the data derived from interactions of users while browsing the web. Web usage data includes the data from web server access logs, proxy server logs, browser logs, user profiles, registration data, cookies, user queries etc. A web server log is an important source for performing web usage mining because it explicitly records the browsing behavior of site visitors. The typical problem (solved in the data preprocessing step) is thus distinguishing among unique users, server sessions episodes etc.

Web usage mining focuses on techniques that could predict user behavior while the user interacts with the web. The applications of web usage mining could be classified into two main categories: (1) learning user profile or user modeling, and (2) learning user navigation patterns [10]. The methods used for web usage mining are (descriptive) statistical analysis, association rules (to relate pages that are often referenced together), clustering (to build usage clusters or page clusters), classification (to create user profiles), sequential pattern discovery (to find inter-session patterns such that the presence of a set of page views is followed by another set of page views in a time-ordered set of episodes), or dependency modeling (to capture significant dependencies among various variables in the web domain) [13]. Some systems already exist in this area: WebSIFT (uses clustering, statistical analysis and association rules) [4], WUM (looks for association rules using an extended version of SQL) [12], or WebLogMiner (combines OLAP and KDD) [17]. An overview of web page recommendation systems is presented in [7].

The algorithms described in this paper are motivated by two goals – (1) to predict user’s behavior (i.e. to recommend the next page of interest given previously visited pages), and (2) to find interesting patterns in the visited page sequences. In the following we describe predictors of the next page type visited (e.g. product detail, FAQ, advice) and of the next product type of interest (e.g. digital cameras or zoom lenses). Both predictors can be used by web servers to recommend to users where they could go next. For practical purposes, such predictors might produce several recommendations so that the user spots an interesting page with greater probability. On the other hand, interesting patterns identified in page sequences provide important information both to webmasters and marketing specialists – e.g. about two products often bought together, or about a product detail page often being followed by a visit to a FAQ page. We present two types of algorithms – one statistical, which is appropriate for the next page prediction task, and two rule-based, which are suitable for both page prediction and pattern discovery.

In section 2 we list basic statistics of the data. Section 3 presents results of a statistical Markov N-gram page predictor. Section 4 describes two rule-based approaches, based on a set covering algorithm and a compositional algorithm. Section 5 compares the above and sections 6 and 7 conclude with directions for future research.

2 Click-Stream Data

The analyzed data comes from a Czech company that operates several e-shops. The log file (about 3 millions of records – the traffic of 24 days) contained the usual information: time, IP address, page request URL and referer. In addition to this, the log data contained also a generated session ID so the identification of users was relatively easy (see Fig. 1) – we treated each sequence of pages with the same ID as one session.

```

unix time; IP address;          session ID;          page request; referee;
1074589200; 193.179.144.2; 1993441e8a0a4d7a4407ed9554b64ed1; /dp/?id=124; www.google.cz;
1074589201; 194.213.35.234; 3995b2c0599f1782e2b40582823b1c94; /dp/?id=182;
1074589202; 194.138.39.56; 2fd3213f2edaf82b27562d28a2a747aa; /; http://www.seznam.cz;
1074589233; 193.179.144.2; 1993441e8a0a4d7a4407ed9554b64ed1; /dp/?id=148; /dp/?id=124;
1074589245; 193.179.144.2; 1993441e8a0a4d7a4407ed9554b64ed1; /sb/; /dp/?id=148;
1074589248; 194.138.39.56; 2fd3213f2edaf82b27562d28a2a747aa; /contacts/; /;
1074589290; 193.179.144.2; 1993441e8a0a4d7a4407ed9554b64ed1; /sb/; /sb/;

```

Fig. 1. Part of the web log

The whole dataset consists of 522,410 sessions, of which 318,523 only contain a single page visit. In the following, we will reduce our dataset to the 203,887 sessions that contain at least two page visits. The average length of these sessions was 16; the median was 8 and modus 2. A session length histogram is shown in Fig. 2. For evaluation purposes, we split the 203,887 sessions into three sets: a

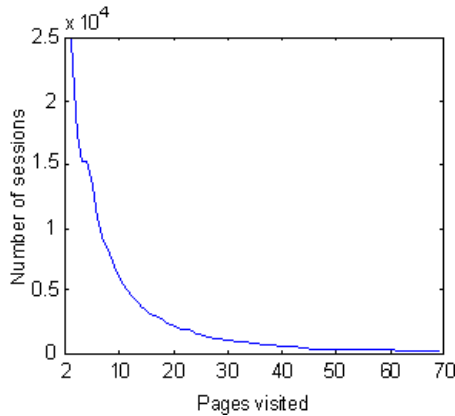


Fig. 2. Session length histogram

training set (the first 100,000 sessions¹), a test set (the second 60,000 sessions) and a held-out set (the remaining 43,887 sessions).

In addition to the log data, we had the following information: table `shop` listed all e-shops (7 entries), table `category` listed general product categories (65 entries), table `product` contained product subcategories (157 entries) and table `brand` listed all sold brands (197 entries).

Each page request had the same structure: `page_type/content_ID`. This allowed us to identify two interesting types of information in the click-streams: sequences of page types (e.g. detail of a product, shopping cart, product comparison), and sequences of product categories (both the `product` and `category` tables combined). For page types, we appended an extra 'end' page to the end of each page sequence, since it seemed important to model which page sequences lead to leaving the web. There were 22 distinct page types including the 'end' page. The majority class was the listing of products (37.7%) followed by a detail view of a product (31.3%). For product categories, no 'end' pages were appended. Based on the `product` and `category` tables, we defined 32 new product categories that grouped similar entries from both tables and used these to define the product sequences. The majority class were cameras (13.1%) followed by video & DVD (10.3%). All pages that did not contain product category information were removed from product sequences. Subsequent occurrences of the same product category were replaced by a single occurrence. The average length of the resulting product sequences was 2.

3 Markov N-Gram Predictor

Statistical approaches to modeling sequences are widely used in areas such as speech recognition, natural language processing or bio-informatics. As our first approach, we trained various *Markov N-gram models* (N-gram models for short)

¹ Sessions were sorted by session start time.

from click-stream sequences. Using these models, we predicted the most likely page following after each history in test data. A model similar to our implementation is described in [6].

3.1 N-Gram Model

An N-gram model models the probability of an observed page sequence $A_1A_2 \dots A_n$ as:

$$P(A_1A_2 \dots A_n) = \prod_{i=1}^n P(A_i|A_{i-k} \dots A_{i-1}) \quad (1)$$

where k is the length of history taken into account. A model with k -token histories is called a $(k + 1)$ -gram model and obeys the Markov property of limited memory and stationarity [8].

An important decision is how to model the probability $P(A_i|A_{i-k} \dots A_{i-1})$. Based on counts observed in training data, a k -gram probability P_k only conditioned on $(k - 1)$ -length histories can be estimated as

$$P_k(A_i = c|A_{i-k+1} = a \dots A_{i-1} = b) = \frac{n(a \dots bc)}{n(a \dots b)} \quad (2)$$

where $n(xy)$ is the count of page sequence xy observed in training data. However, choosing just one fixed history length is problematic since long histories suffer from data sparsity and short histories may not contain sufficient information about the next page. It is therefore a common approach to interpolate P_k with lower-order probabilities (i.e. those conditioned on shorter histories). The interpolated (smoothed) k -gram distribution is thus given as a weighted sum of the fixed history length distributions:

$$P(A_i|A_{i-k+1} \dots A_{i-1}) = \lambda_0 P_0(A_i) + \lambda_1 P_1(A_i) + \lambda_2 P_2(A_i|A_{i-1}) + \dots + \lambda_k P_k(A_i|A_{i-k+1} \dots A_{i-1}) \quad (3)$$

where

$$\sum_{i=0}^k \lambda_i = 1, \forall_{i=0}^k \lambda_i \geq 0, \lambda_i \leq 1 \quad (4)$$

In Eq. [3] P_0 is a uniform distribution over all existing pages, P_1 is the unigram distribution only based on page frequencies without taking history into account, and the remaining P_i distributions are conditioned on histories of length $i - 1$.

The weights λ_i are best determined by an unsupervised EM algorithm [8], which iteratively re-estimates weights using held-out data until convergence. First, all the components P_i ($i = 1 \dots k$) are computed from counts observed in the *training dataset* according to Eq. [2]. Second, weights are set to some initial non-zero (we chose uniform) values. Third, the probability of a *held-out dataset* is computed using the interpolated distribution in Eq. [3] with the working weights. For each addend in Eq. [3], its share on the total held-out data probability is collected. After running through all held-out data, the new weights are obtained by normalizing the shares of each addend. Step three is repeated until none of the λ_i weights changes significantly.

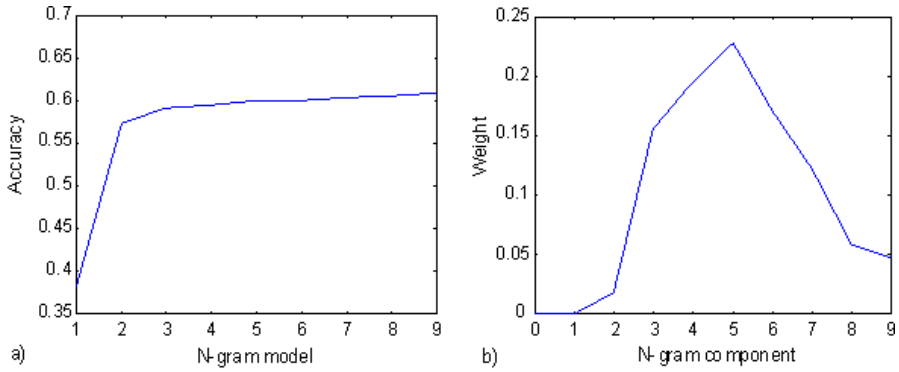


Fig. 3. a. N-gram accuracy for page types b. Weights of 9-gram components

3.2 N-Gram Results

We trained N-gram models to predict the next page type, and the next product category a user might be interested in.

All N-gram models are only trained using the training data set, and smoothed using the held-out data set. We evaluate the N-gram predictors by comparing the real item in test data with the most likely item that would follow the observed history according to the model. Prediction accuracy is given as the number of correct predictions over all predictions. The best results are compared to rule-based methods in Table 4. The table also includes results achieved on training data; in these cases, N-gram model weights were estimated on training data as well, which caused the highest-order component to always have a weight of 1.

Accuracy for page types reached 0.61 and is reported in detail in Fig. 3a for N-gram models with N ranging from 1 (a unigram model only smoothed with uniform distribution) to 9 (a 9-gram model smoothed with 9 lower order distributions). We observe that accuracy climbs significantly until the trigram model, thus the chosen page type depends mostly on the two preceding pages.

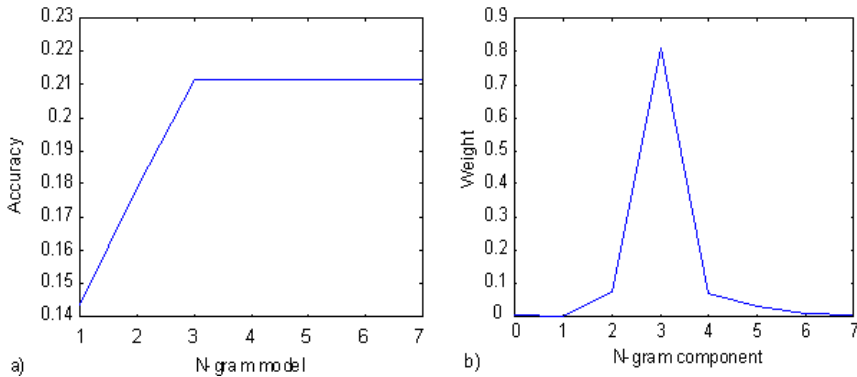


Fig. 4. a. N-gram accuracy for product types b. Weights of 7-gram components

It is also interesting to note the smoothing weights for the 9-gram model in Fig. 3b. Here, the pentagram component’s weight reaches a maximum of 23% although its contribution to the overall accuracy is small. We assume this is due to abundance of data – data sparseness is not significant yet for pentagrams and thus we can get large weights for high-order probabilities.

Accuracy for product types reached 0.21 for a heptagram model; climbing from 0.14 for a unigram model, as seen in Fig. 4a. Here, we observe a significant increase in accuracy until the trigram model, where further increases stop. This is confirmed by the superior trigram component weight in Fig. 4b. We can thus conclude that the next product type of interest can be reasonably predicted based just on two preceding products.

We performed additional experiments with both page and product types to find out how the prediction accuracy varies with the amount of available training data. For this purpose we trained 5-gram models from training sets of the following sizes (in sessions): 20, 50, 100, 200, 500, 1k, 2k, 5k, 10k, 20k, 50k and the full 100k training sessions. All data sets were constructed by taking the first K sessions from the full training set. For each model, its N-gram component weights were re-estimated on the held-out set. Accuracies of these models were measured on the test set and are shown in Fig. 5.

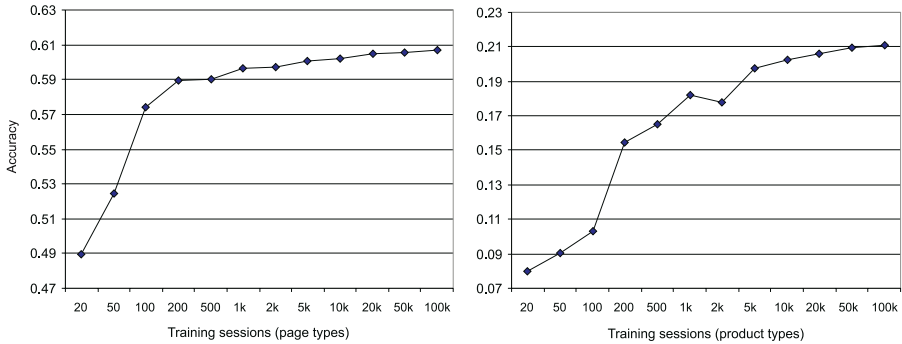


Fig. 5. 5-gram model accuracies for reduced training data sizes

For both page and product types, we observe that the model accuracies climb steeply even for surprisingly small training data sets. In case of product types the climb is less steep since the product sequences are shorter (average length is 2 compared to 16 for page types) and thus the same amount of sessions contains less training data.

The 5-gram component weights were also monitored for each of the trained models. Fig. 6 confirms that larger training data leads to better estimates of the N-gram components. We observe that when trained on larger data, the re-estimation procedure distributes more weight to the higher order components since they better fit the held-out data. The graph shows cumulative weights starting from the uniform component (0), unigram (1), up to the 5-gram.

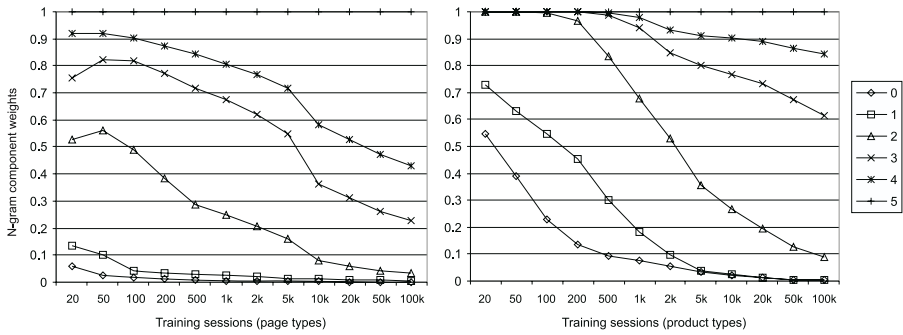


Fig. 6. Development of 5-gram component weights for reduced training data sizes

4 Rule-Based Predictors

4.1 Classical Rule Learning Algorithms

Decision rules in the form

$$Ant \Rightarrow Class$$

where *Ant* (antecedent, condition) is a conjunction of values of input attributes (called categories or selectors) and *Class* is a category of class attribute *C*, are one of most popular formalisms how to express classification models learned from data. The commonly used approach to learning decision rules is the *set covering approach* also called “separate and conquer”. The basic idea of this approach is to create a rule that covers some examples of a given class and remove these examples form the training set. This is repeated for all examples not covered so far. There are two basic ways how to create a single rule:

1. by rule generalization, i.e. by removing categories from antecedent of a potential rule (starting from a rule with categories of all input attributes in the antecedent) - this method is used in the AQ algorithms by Michalski (see e.g. [11]).
2. by rule specialization, i.e. by adding categories to the antecedent of a potential rule (starting from a rule with empty antecedent) – this method is used e.g. in CN2 [5] or CN4 [3].

The other way how to create decision rules is the *compositional approach*. In this approach the covered examples are not removed during learning, so an example can be covered with more rules. Thus more rules can be used during classification. In compositional approach, all applicable rules are used and their particular contributions to classification are combined into the final decision. To do this, some numerical value is usually added to the rule, the simplest one is the rule confidence (also called validity) defined as $n(Ant \wedge Class) / n(Ant)$, where $n(Ant \wedge Class)$ is the number of examples that match both *Ant* and *Class* and $n(Ant)$ is the number of examples that match *Ant* in the data.

4.2 Rule Learning Algorithms for Click-Streams

The main difference to conventional rule learning algorithms is due to the fact that instead of unordered set of categories we deal with an ordered sequence of pages. So we are looking for rules in the form

$$Ant \Rightarrow page(p)$$

where Ant is a sequence of pages, $page$ is a page view that directly follows the sequence Ant , and p is the validity of the rule

$$p = \frac{n(Ant//page)}{n(Ant)}.$$

In the formula above we denote the number of occurrences of a sequence in the data by $n(sequence)$ and a concatenation of two sequences $s1$ and $s2$ by $s1//s2$.

We propose two rule learning algorithms for click-streams: a set covering and a compositional one [2]. We follow the rule learning approach based on rule specialization in our algorithms as well. As we assume that most relevant for prediction of occurrence of a page are pages that are closest to this page, the specialization of the rule $Ant \Rightarrow page$ is done by adding a new page to the beginning of the sequence Ant . Analogously, a generalization of the rule $Ant \Rightarrow page$ is done by removing a page from the beginning of the sequence Ant .

The main idea of our *set covering* algorithm is to add (for a particular page to be predicted) rules of growing length of Ant . We check each rule against its *generalization* created so far. Adding a new rule $Ant \Rightarrow page$ to the model is determined by χ^2 test that compares the validity of these two rules. If the rule in question is added to the model, its *generalization is updated* by re-computing the validity by ignoring (removing) sequences that are covered by the newly added rule (Fig. 7).

The main idea of our *compositional* algorithm (Fig. 8) is again to add (for a particular page to be predicted) rules of growing length of Ant . We check each rule against the *results of classification* done by all rules created so far. Adding a new rule $Ant \Rightarrow page$ to the model is determined by χ^2 test that compares its validity p with the weight of predicted page inferred during classification for the sequence Ant . The weight $w^\oplus(Ant)$ of predicted page is computed according to the formula

$$w_1 \oplus w_2 = \frac{w_1 \times w_2}{w_1 \times w_2 + (1 - w_1) \times (1 - w_2)}, \quad (5)$$

w_1 and w_2 in this formula denote weights of rules that are applicable to the sequence Ant .

4.3 Rule Learning Algorithm Results

In the first set of experiments we were looking for rules that can be interpreted as interesting by the data providers. We identified as interesting e.g. the rules

Initialization

for each page occurring in the data

1. compute its relative frequency in the data as $P = (\text{no. of occurrences of } page \text{ in the input episodes}) / (\text{no. of all input episodes})$
2. if $P \geq n_{min}$
 - 2.1 add $default \Rightarrow page$ into the list of rules *Rules*
 - 2.2 add $page$ into list of pages *Pages*
 - 2.3 add $default \Rightarrow page$ into list of implications *Impl*

Main loop

while *Impl* not empty do

1. take first rule $Ant \Rightarrow page$ from *Impl*
2. if length of $Ant < l_{max}$ then
 - 2.1 for each page pp from *Pages*
 - 2.1.1 find the most specific generalization of the rule $pp // Ant \Rightarrow page$ in *Rules* (denote it $AntX \Rightarrow page$)
 - 2.1.2 compare (using chi2 test) the validity of rules $pp // Ant \Rightarrow page$ and $AntX \Rightarrow page$
 - 2.2 from all created rules $pp // Ant \Rightarrow page$ select the one with the most significant difference in validity (denote this rule $pp_{best} // Ant \Rightarrow page$)
 - 2.3 if $pp_{best} // Ant \Rightarrow page$ significantly at a given significance level differs from $AntX \Rightarrow page$ then
 - 2.3.1 add rule $pp_{best} // Ant \Rightarrow page$ to *Rules* and *Impl*
 - 2.3.2 re-compute the validity of rule $AntX \Rightarrow page$ by taking into account only episodes containing $AntX$ and not containing Ant
 - 2.3.3 recursively update *Rules* (i.e. find the most specific generalization of $AntX \Rightarrow page$, compare this generalization with $AntX \Rightarrow page$, remove $AntX \Rightarrow page$ from *Rules* if the difference is not significant etc.)
3. remove $Ant \Rightarrow page$ from *Impl*

Fig. 7. The set covering rule learning algorithm for click-stream analysis

```
dp, sb -> sb (Ant: 5174; AntPage: 4801; P: 0.93)
ct -> end (Ant: 5502; AntPage: 1759; P: 0.32)
faq -> help (Ant: 594; AntPage: 127; P: 0.21).
```

for the sequences concerning page types. In the listing above, *ct* stands for "contact", *Ant* stands for $n(Ant)$ and *AntPage* stands for $n(Ant // Page)$. Among rules concerning product categories we found e.g.

```
loud-speakers -> video + DVD (Ant: 14840, AntPage: 3785, P: 0.26)
data cables -> telephones (Ant: 2560, AntPage: 565, P: 0.22)
PC peripherals -> telephones (Ant: 8671, AntPage: 1823, P: 0.21)
```

The obtained rule sets can directly be used to predict the behavior of a user. So e.g. for a sequence of pages *dp*, *sb* the system will predict *sb* as the next page, and for the sequence *loud-speakers* the system will predict *video + DVD*.

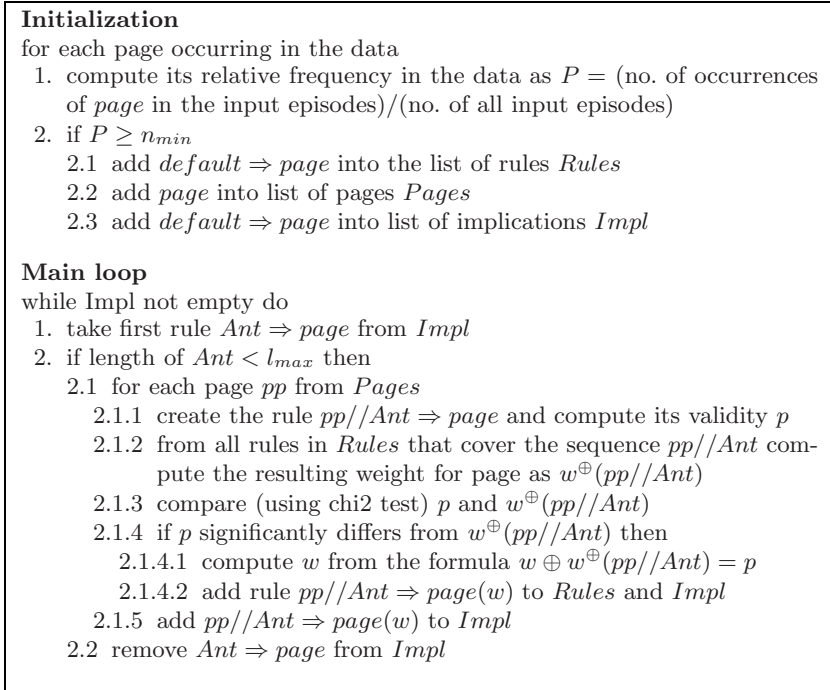


Fig. 8. The compositional rule learning algorithm for click-stream analysis

In the second set of experiments we were interested in classification accuracy of our rule sets. We have run our algorithms repeatedly for both page sequences (first set of experiments) and product sequences (second set of experiments). When looking at the differences between the set covering and compositional algorithms, we can observe different trade offs between comprehensibility and accuracy: the set covering algorithm offers higher accuracy but lower comprehensibility (larger number of rules) than the compositional algorithm (see Tab. [II](#)). The default accuracy refers to the accuracy of the “zero rule” model, that always predicts the most frequent page. In all experiments we exceeded this “base line”. Since both rule-based methods make no use of held-out data, this data was appended to the training data in all cases.

5 A Comparison of N-Gram and Rule-Based Models

In Table [II](#), accuracy is computed as the number of correct guesses according to the model and to the observed history, divided by the total number of guesses (all pages from the evaluated data). For the N-gram model, the predicted page is the one with the highest conditional probability given an observed history of pages. For both rule based methods, the predicted page is the page for which the combined weight of all applicable rules (given history) is maximal.

Table 1. Empirical comparison of algorithms

page sequences	default ^a	N-gram accuracy	set covering		compositional	
	accuracy		no.rules	accuracy	no.rules	accuracy
test data	0.40	0.61	1088	0.59	371	0.49
training data	0.40	0.67	1088	0.60	371	0.50
product sequences						
test data	0.14	0.21	2176	0.24	903	0.19
training data	0.14	0.24	2176	0.23	903	0.18

^a Default accuracy corresponds to a model that will always predict the most frequent page, i.e. it is the relative frequency of the majority page.

We observe that for both page and product types, the accuracy of the N-gram models on test data is comparable to the set covering algorithm. For page types, the best prediction accuracy of 0.61 was reached by a 9-gram model, while the set covering algorithm achieved the highest accuracy of 0.24 for product types. When applied as a page recommendation system, both predictors should yield more than one recommendation about the next page to let the user choose from several relevant pages. In terms of performance on training data, the 9-gram model achieves 0.67 accuracy, which is however caused by over-fitting the training data. On the other hand, both rule-based algorithms seem to be resistant to over-fitting, surprisingly for product sequences they achieve slightly better results on test data than on the training set.

To compare N-gram and both rule based methods, we may view N-gram models as exhaustive sets of weighted rules. Each non-zero probability $P_i(c|a\dots b)$ from Eq. 2, weighted by the corresponding weight λ_i , can be treated as confidence of a corresponding rule $a\dots b \Rightarrow c$. For a particular history $a\dots b$, the weighted confidences of relevant rules are summed to produce a probability for each possible predicted class. This similarity in learnt models seems to be the reason why the set-covering algorithm with large amounts of rules performed comparably to the N-gram models.

Viewing N-gram models as sets of rules, there are however several major distinctions from the rule-based algorithms. First, the number of N-gram “rules” is exhaustive, although we could e.g. remove all “rules” conditioned on histories having count less than a chosen threshold. Second, the contributions of N-gram “rules” are based on confidence (as for the set covering algorithm), however they are further weighted by a constant factor that expresses the reliability of all rules with a certain length of antecedent. To relax this constraint, N-gram weights could be alternatively specified for intervals of history frequencies (referred to as bucketed smoothing) instead of history lengths; in this case frequent histories would receive higher weights as they are more reliable. Yet another difference lies in the method how contributions of multiple rules are combined. In the compositional approach, Eq. 5 is used, whereas a weighted sum in Eq. 3 is used for the N-gram model to yield probability. Last of all, unlike the set covering algorithm, in the N-gram case multiple rules are used during a single classification, as in the compositional approach.

When the analysis goal is user's understanding of learnt models (rather than prediction), a method which learns few comprehensive rules is generally preferable. In our case, this is the compositional approach. The learnt weights of N-gram model components also seem to contain useful information on how long histories still influence user's behavior.

6 Future Work

For the N-gram predictor, we plan to implement other smoothing methods, starting with bucketed smoothing based on history frequencies. For the rule-based models, we will analyze how rule learning parameters impact accuracy and compare our algorithms with state-of-the-art methods. Another experiment we would like to perform on the described dataset is predicting the next page based on a richer set of features observed in history. These features would include page types, visited product types, product makes, session length, and the time spent on these pages. A suitable prediction algorithm that would benefit from a large feature set could be a Maximum Entropy Model, which was successfully applied in a web recommendation system described in [9].

7 Conclusion

We presented and compared two approaches for clickstream data analysis – one statistical and two rule-based algorithms. Using these algorithms, we predicted the next page type visited and the next product type of interest. The statistical N-gram algorithm and the set-covering rule-based algorithm achieved comparable prediction accuracies for both page and product types. On the other hand, the compositional rule-based algorithm, which was inferior in terms of prediction accuracy, proved to be suitable for discovering interesting patterns in page sequences. The described algorithms can be applied by web servers to recommend relevant pages to their users, and to identify interesting patterns in their log files.

Acknowledgements

The research is supported by the grant no.MSM138439910 of the Ministry of Education of the Czech Republic and the grant no.201/05/0325 of the Czech Science Foundation.

References

1. Berka, P., Ivánek, J.: Automated knowledge acquisition for PROSPECTOR-like expert systems. In: Bergadano, F., De Raedt, L. (eds.) Machine Learning: ECML-94. LNCS, vol. 784, pp. 339–342. Springer, Heidelberg (1994)
2. Berka, P., Laš, V., Kočka, T.: Rule induction for click-stream analysis: set covering and compositional approach. In: IIPMW 2005. LNCS, pp. 13–22. Springer, Heidelberg (2005)

3. Bruha, I., Kočková, S.: A support for decision making: Cost-sensitive learning system. *Artificial Intelligence in Medicine* 6, 67–82 (1994)
4. Cooley, R., Tan, P.N., Srivastava, J.: Discovery of interesting usage patterns from web data. Tech. Rep. TR 99-022, Univ. of Minnesota (1999)
5. Clark, P., Niblett, T.: The CN2 induction algorithm. *Machine Learning* 3, 261–283 (1989)
6. Deshpande, M., Karypis, G.: Selective Markov Models for Predicting Web-Page Accesses. Technical Report 56, University of Minnesota (2000)
7. Gündüz, S., Özsü, M.T.: Recommendation Models for User Accesses to Web Pages. In: Kaynak, O., Alpaydm, E., Oja, E., Xu, L. (eds.) *ICANN 2003 and ICONIP 2003*. LNCS, vol. 2714, Springer, Heidelberg (2003)
8. Jelinek, F.: *Statistical Methods for Speech Recognition*. MIT Press, Cambridge (1998)
9. Jin, X., Mobasher, B., Zhou, Y.: A Web Recommendation System Based on Maximum Entropy. In: *Proc. IEEE International Conference on Information Technology Coding and Computing, Las Vegas* (2005)
10. Kosala, R., Blockeel, H.: Web Mining Research: A Survey. In: *SIGKDD Explorations*, vol. 2(1) (2000)
11. Michalski, R.S.: On the Quasi-minimal solution of the general covering problem. In: *Proc. 5th Int. Symposium on Information Processing FCIP'69, Bled*, pp. 125–128 (1969)
12. Spiliopoulou, M., Faulstich, L.: WUM: A tool for web utilization analysis. In: Atzeni, P., Mendelzon, A.O., Mecca, G. (eds.) *The World Wide Web and Databases*. LNCS, vol. 1590, Springer, Heidelberg (1999)
13. Srivastava, J., Cooley, R., Deshpande, M., Tan, P.N.: Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations* 1(2) (2000)
14. Witten, I.H., Frank, E.: Generating Accurate Rule Sets Without Global Optimization. In: *Proc. of the 15th Int. Conference on Machine Learning, Morgan Kaufmann, San Francisco* (1998)
15. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco (1999)
16. Zaiane, O., Han, J.: WebML: Querying the World-Wide Web for resources and knowledge. In: *Workshop on Web Information and Data Management WIDM'98, Bethesda*, pp. 9–12 (1998)
17. Zaiane, O., Xin, M., Han, J.: Discovering web access patterns and trends by applying OLAP and data mining technology on web logs. In: *Advances in Digital Libraries* (1998)

Improved IR in Cohesion Model for Link Detection System

K. Lakshmi and Saswati Mukherjee

Anna University, India

Lakshmi_tamil@hotmail.com, msaswati@yahoo.com

Abstract. Given two stories, Story Link Detection System identifies whether they are discussing the same event. Standard approach in link detection system is to use cosine similarity measure to find whether the two documents are linked. Many researchers applied query expansion technique successfully in link detection system, where models are built from the relevant documents retrieved from the collection using query expansion. In this approach, success depends on the quality of the information retrieval system. In the current research, we propose a new information retrieval system for query expansion that uses intra-cluster similarity of the retrieved documents in addition to the similarity with respect to the query document. Our technique enhances the quality of the retrieval system thus improving the performance of the Link Detection System. Combining this improved IR with our Cohesion Model provides excellent result in link detection. Experimental results confirm the effect of the improved retrieval system in query expansion technique.

Keywords: Topic Detection and Tracking, Story Link Detection System, Information Retrieval System, Cohesion Model.

1 Introduction

Topic Detection and Tracking System is receiving an increased attention in recent days for its application in many areas like information retrieval, question answering system and summarization. In information retrieval systems, users are interested in extracting information about a specific topic. In question answering system, it is necessary to organize the document according to particular topic and search for answer within the topic set. Users may be interested in viewing summarized information about certain topic. Therefore organizing information on the basis of topic is important in many applications [1] [2].

TDT consists of five sub tasks: Story Segmentation - Segmenting a stream of data into distinct stories, First Story Detection - Identifying those news stories that are the first to discuss a new event occurring in the news, Cluster Detection - Given a small number of sample news stories about an event, finding all following stories in the stream, Tracking - Monitoring the news stream for finding additional incoming stories that can be added to the existing topics, and Link Detection – Deciding whether any two randomly selected stories discuss the same topic [3].

Out of the five subtasks, Link detection system is considered to be the core component of TDT, as it can be used for performing all the other tasks as well. This paper deals with the link detection system. Task in hand is to identify whether the two given stories talks about the same event, where an event is something that happens at a specific place and time and involves specific participants, be they human or otherwise [4].

1.1 Proposed Model

Encouraged by the good performance of the query expansion technique in link detection system, we propose to use an improved query expansion model. Performance of such systems depends on the quality of the retrieved documents. Given two documents D_1 and D_2 as input to the link detection system, each document is considered as query and documents that are relevant to the query are retrieved from the collection. The given documents D_1 and D_2 are named as query documents and will be referred as such in the following discussion. From the relevant documents of each query document D_k , corresponding model M_k is built. Information retrieval systems retrieve not only relevant documents but also non-relevant documents. Worse is, many a time, negative documents are classified as relevant documents. When a model M_k is built out of relevant as well as non-relevant documents, many terms that are not relevant to the original query document D_k will be placed in the model M_k . Under such circumstances, M_k is not the true representative of D_k . To avoid this problem, we propose an improved information retrieval system that filters the retrieved documents through one additional level, so that only relevant documents are obtained. This additional level of filtering is performed based on the intra similarity, thereby improving the quality of the overall information retrieval system and reducing the chances of retrieving non-relevant documents. This, therefore, improves the quality of the model generated from the retrieved relevant set.

Reference [14] has used cohesion model in story link detection system, where query expansion has been employed using the concepts of cohesion technique. In this approach, mechanism of model building exploits the terms' individual contribution and its distribution in the relevant documents. To achieve this, four factors, *viz.*, Sense of belonging, Feelings of morale, Goal consensus, and Network cohesion have been used. This approach has produced improved result over the basic model. Our proposed model combines the improved IR along with cohesion model.

Once the models are built for each of the query documents, they are compared using Modified Fraction similarity [10]. Modified Fractional Similarity measure gives credit for the overlapping term and reduces the similarity score for having non-overlapping terms. Hence the similarity score in this method depends on both similar as well as non-similar terms. With the encouraging performance in text classification, we use Modified Fractional Similarity to compare the models in our link detection system. Query documents D_1 and D_2 are declared as linked if the similarity measure between the models is greater than a predefined threshold.

We have organized this paper with section 2 discussing about the related work, section 3 elaborates the proposed system, section 4 about the experiment results and the next section discusses further possible enhancements.

2 Related Works

A number of research groups have developed story link detection systems. The basic model of link detection system proposed by [7] used vector space model to represent the given documents D_1 and D_2 . Then D_1 and D_2 are compared using cosine similarity. This method is a well-established method that produces consistent result in various tasks and data set.

Reference [5] proposed a model that uses source-pair information and various similarity measures of document pairs for training the support vector machine. As the documents come from different sources like broadcast and newswire, each has different characteristics. Similarity measure between broadcast-broadcast, newswire-newswire and broadcast-newswire are different. They have captured this information and shown that inclusion of source-pair information helps to improve the link detection.

Reference [11] has used the concept of query expansion, a well-established technique in IR. Here document given for link detection is considered as query and the documents that are relevant are retrieved from the collection. By using local context analysis technique, terms in the relevant documents are added to the query document. Thus each document is expanded and then compared. This technique showed slight improvement over the link detection system that does not use any query expansion. However success of the model depends on how successfully relevant documents alone are fetched and how the terms are assigned weights. In their paper they have indicated that expanding document using non-relevant document set would move the model in the wrong direction and would severely affects the performance of the link detection system.

Reference [6] has used probability method for obtaining the relevant documents, *i.e.*, probability of a document to be retrieved, for the given query $P(D|Q)$, is calculated and then topic model is built for each given story (query). Each term in the relevant documents are assigned weight according to the term's probability in the document $P(w|D)$ and the probability of the document to be retrieved for the given query $P(D|Q)$. The two topic models thus obtained are compared by using Clarity adjusted Kullback-Leibler divergence method.

In the language models terms in a document are considered to be independent of each other, which is not true in the real case. Reference [9] has exploited term dependency to capture the underlying semantics in the document. They proposed modeling sentences, rather than words or phrases as individual entities.

Using "soundex" in comparing documents of different sources (broadcast and newswire) is proposed by [13]. When broadcast news is converted to text, most of the nouns are given different spelling. So when a broadcast news and newswire news are compared, some of the terms do not match and this may lead to low similarity value. System proposed by [13] addressed this problem. However, they were not able to produce better result due to poor named-entity recognizer for ASR documents.

3 Cohesion Model with Improved IR

Documents relevant to the query documents are retrieved using the proposed improved IR. Document D_k is expanded using the terms in the relevant documents.

Then each term is assigned weight according to social cohesion concept. The expanded models are compared to decide whether the given two documents are linked.

For comparing the documents, we have used Modified Fractional Similarity measure proposed in [10]. Modified Fractional Similarity is given in equation (1). In this phase of link detection, it is employed to compare the query document with each of the documents in the collection. The choice of this similarity measure here has stemmed from the fact that this similarity technique provides better precision and reduces false positives in the overall retrieved set.

First step of the process is to get the relevant documents considering the given document as query and the second step is to assign weight to the terms of relevant documents.

3.1 Relevant Documents

For expanding the query documents, terms in the retrieved relevant document set are used. In this context, it is important that we consider only the relevant documents, since false positives may move the model into a wrong direction. Thus improved IR technique having very high accuracy is the need of the hour. To obtain this high precision, we propose to obtain the set of relevant documents in two steps. First the relevant documents are obtained using IR and then this set is filtered further to take out any noncontributing documents that may have been retrieved.

$$\begin{aligned}
 \text{Mfraction}(d1,d2) &= \frac{2 * \alpha}{\beta + \gamma} && \text{if } \{d1\} - \{d2\} \neq \varnothing \\
 &= \alpha && \text{if } \{d1\} - \{d2\} = \varnothing \\
 \alpha &= \sum_{i=1}^p w_i * v_i && \text{if } \text{term}_i \in d1 \text{ and } \in d2 \\
 \beta &= \sum_{i=1}^m w_i && \text{if } \text{term}_i \in d1 \text{ and } \notin d2 \\
 \gamma &= \sum_{i=1}^n v_i && \text{if } \text{term}_i \in d2 \text{ and } \notin d1
 \end{aligned} \tag{1}$$

w_i – weight of term_i in $d1$

v_i – weight of term_i in document $d2$

m – number of terms in the $d1$

n – number of terms in the document $d2$

p – number of terms in both $d1$ and $d2$

As the first step, modified fractional similarity (equation 1) is used to retrieve documents those are relevant to the query document D_k . For this purpose, we use, as collection, documents that are temporally closer to the query document. This reduces the search time. We propose to use n previous days documents as our background collection for our first level IR. This period of n days is known as the deferral period.

Our observation is that the consideration of only those documents within the deferral period is sufficient since the events are time specific.

We set a threshold experimentally. Documents that are greater than this predefined threshold are considered as relevant documents.

As shown in Fig. 1, documents R_1 - R_n are relevant documents obtained.

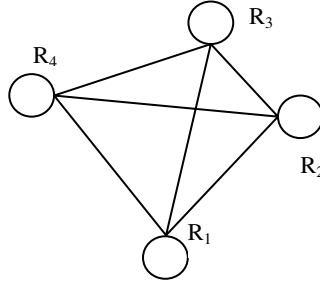


Fig. 1. R_1 - R_4 are the relevant documents. Edges shows connectivity between documents in the relevant document set.

Fig. 1 shows the ideal case of connectivity among the retrieved documents, *i.e.*, all the retrieved documents are connected to each other. In other words all the relevant documents that are obtained in step 1 are similar to each other. But this may not be always true. To ensure minimum connectivity among the retrieved documents, step 2 is implemented as the next filtering step. For this purpose, we use intra-cluster similarity. In this step, similarity of each retrieved document is further measured with respect to other retrieved documents using equation 2 and the relevance is judged on the basis of this measure.

$$c(R_i) = \frac{\sum_{R_j \in \{RL\}} s(R_i, R_j)}{[(n-1)]} \tag{2}$$

Where

$c(R_i)$ – cluster similarity

$\{RL\}$ – Relevant documents set

$s(R_i, R_j)$ – Cosine similarity between R_i and R_j

n – no of relevant documents $|RL|$

Equation 2 uses cosine similarity given by equation 3. Cosine similarity value for two completely relevant documents is 1. Thus the assumption in equation (2) is that, if the document is a true relevant document, its intra similarity (connectivity to other documents) should be equal to $(n-1)$, *i.e.*, equal to the total number of possible edges from this node as shown in the Fig.1. Thus the maximum value for cluster similarity for a document is 1. Document with high connectivity is considered to be more relevant to the given query. For each relevant document, connectivity is measured as the ratio of sum of all similarity measures with respect to other relevant documents and the maximum possible connections (*i.e.*, the number of edges).

$$\text{Cos}(d1,d2) = \frac{\sum d1.d2}{|d1||d2|} \quad (3)$$

The final similarity measure $\text{FS}(D_k, R_i)$ between query document D_k and the relevant document R_i is calculated using equation 4. If the final similarity is greater than a threshold t specified a priori, R_i is considered to be relevant to the given query.

$$\text{FS}(D_k, R_i) = \alpha * s(D_k, R_i) + (1 - \alpha) * c(R_i) \quad (4)$$

Where

$\text{FS}(D_k, R_i)$ – Final Similarity between two documents D_k and R_i ,

D_k is the query document and R_i is the relevant document, $R_i \in RL$

α - Control parameter

$s(D_k, R_i)$ – similarity between D_k and R_i

$c(R_i)$ – cluster similarity

Equation 4 is a weighted sum of $s(D_k, R_i)$ and $c(R_i)$, where $s(D_k, R_i)$ is calculated by equation 1 and $c(R_i)$ is calculated using equation 2. Control parameter α is the used to decide the importance between the two values.

Next step in the process is to assign weight to the terms. Following sub-section explains how the terms are assigned weight according to cohesion-model.

3.2 Building Cohesion-Model

Reference [12] describes an iterative process where “community dialogue” and “collective action” work together to produce social change in a community that improves the health and welfare of all of its members. It discusses Social Cohesion to be one of the factors affecting the Social Change. Social cohesion is an important antecedent and consequence of successful collective action. In [14], this has been used as the base and the corresponding mechanism of building a query expansion model using cohesion factors has been employed.

Since social cohesion consists of the forces that act on members of a group or community to remain in, and actively contribute to the group, a direct mapping of the terms of the relevant documents has been established in [14] to the members in the society. In the current research, we use the term’s contribution and the relationship among them in the relevant documents. Terms in model M_k are assigned weights according to different social cohesion factors.

There are four factors that has been used in the cohesion model for the purpose of Link Detection System. These are:

- Sense of belonging
- Feelings of morale,
- Goal consensus,
- Network cohesion.

Sense of belonging - is the extent to which individual members feel as if they are an important part of the group or community. This can be directly mapped with the term’s frequency in the relevant documents.

$$tf(t) = [1/N] * \sum_{d \in rl} tf(t,d) \tag{5}$$

Where

tf(t) = term frequency of terms in relevant documents

rl – relevant documents

tf(t,d) = term frequency of term t in document d

N – Total no of relevant documents

Feelings of morale – This refers to the extent to which members of a group or community are happy and proud of being a member. We can map this to the *inverse document frequency (idf)* factor of the term. Presence of a term in all or most of the documents across the boundaries of groups in the collection can be viewed as lack of confidence and lack of enthusiasm to identify itself with the group.

Goal consensus – It is the degree to which members of the community agree on the objectives to be achieved by the group. Here we translate it as how many times each term in the relevant document set is repeated in relevant documents. We calculate the *document frequency (df)* of the term in the relevant document set using equation 6. More number of times the term is repeated more overlap is expected among the relevant document set.

$$Df(t) = [1/N] * docfreq(t) \tag{6}$$

Where

df(t) =document frequency of terms t

docfreq –no.of documents term t appears in relevant documents

N – Total no of relevant documents

Network cohesion - This can be viewed as the term’s co-occurrence in the relevant documents. By adding co-occurrence weight in calculating the terms weight is expected to eliminate the problem described in [11]. Even though the quality of the retrieval is poor *i.e.*, the retrieved document set contains negative documents, terms of the negative documents may not co-occur with the positive document terms. Equation 7 is used to calculate the co-occurrence weight.

$$Cnet(t_i) = \sum_{t_i, t_j \in \{T\}} [n(t_i \cap t_j)/n(t_i)] * [tf(t_i) / p] \tag{7}$$

Where

T – All terms in the relevant documents

Cnet(t_i) - Cohesion value of term t_i

n(t_i ∩ t_j) – no of times terms t_i & t_j co-occurred in relevant documents

n(t_i) - document frequency of term t_i in relevant documents

p - no.of terms in the relevant documents

tf(t_i) – term frequency of term t_i

With all the parameter final weight of the term is calculated as given in equation 8.

$$w(t_i) = c1 * tf(t_i) + c2 * idf(t_i) + c3 * df(t_i) + c4 * Cnet(t_i) \quad (8)$$

Where

$w(t_i)$ - weight of term t_i

$tf(t_i)$ = term frequency of terms in relevant documents

$idf(t_i)$ = inverse document frequency of term t_i in relevant documents

$df(t_i)$ = document frequency of term t_i in relevant documents

$Cnet(t_i)$ - Cohesion value of term t_i

$c1-c4$ - constants

Each term is given weight as the weighted sum of the tf , idf , df and $Cnet$. Equation 8 shows how the term weight of each term is calculated. Constants $c1-c4$ are selected empirically.

Thus a combined work using improved IR and cohesion model is established in this work.

3.3 Comparing Models

For each query document D_k relevant documents are retrieved. Model M_k , which is a representative of D_k , is built from the terms of the relevant documents. In M_k terms are assigned weights according to cohesion mechanism. Two models M_1 and M_2 built for the two query documents are compared using MF Similarity to establish link.

4 Experimental Results

In this section we evaluate performance of Cohesion Model with improved IR on the Link detection task of TDT. First, we describe the experimental setup and the evaluation methodology

4.1 Experimental Set Up

We have used TDT4 data for evaluating our proposed system. We have considered 16 topics' data for the experiment. Test data contains 377 positive links and 1277 negative links. The news stories were collected from different sources; newswire sources (Associated Press and New York Times) and broadcast sources (Voice of America and Public Radio International). We consider only English stories for our experiments. Though the corpus contains other language documents, those are not considered for these experiments. Text version of the broadcast news is used for this evaluation.

4.2 Evaluation Method

The system is evaluated in terms of its ability to detect the pairs of stories that discuss the same topic. During evaluation, the Link Detection System emits a YES or NO decision for each story pair. If our system emits a YES for an off-target pair, we get a False Alarm error; if the system emits a NO for on-target pair, we get a Miss error. Otherwise the system is correct.

Link Detection is generally evaluated in terms of F1-Measure as in classification system or Cost Function given by equation 9, which is the weighted sum of probabilities of getting a Miss and False Alarm [8].

$$\text{Cost} = P(\text{Miss})C_{\text{Miss}} + P(\text{FA})C_{\text{FA}} \quad (9)$$

In the present work, we have considered F1-measure as the main factor for the evaluation of the performance of the various systems that we have used for our experimentation. F1-measure has been chosen because we want to use the system as a basic component of TDT. In [8], Chen *et al.* has shown that optimized story link detection is not equivalent to optimized new event detection. An optimal link detection system tries to reduce the false alarm (as the weight of the false alarm is high). But false alarm of Link Detection system is equivalent to miss in New Event Detection System (NED). Thus an optimized Link Detection System does lead to optimized NED. So we have used F1-measure to indicate the performance of the Link Detection System, as F1-measure is a harmonic mean of precision and recall.

We have considered 14 different Story Link Detection systems for evaluation as given in table 1.

Table 1. Various Link Detection Systems considered for testing

S.No	SLD Systems
1	Tf
2	tf+idf
3	tf+Cnet
4	tf+50*Cnet
5	tf+df
6	tf+idf+50*Cnet
7	tf+idf+Cnet
8	tf+idf+df
9	tf+df+50*Cnet
10	tf+df+Cnet
11	tf+idf+df+50*Cnet
12	tf+idf+df+Cnet
13	tf*idf
14	tf*idf*df

Various factors of social cohesion are considered in addition to the base factor sense of belonging, which is represented using *term frequency (tf)*. Then one by one, other factors are added with different weights to evaluate their contribution in the system performance. Systems 1-12 consider linear combination of these factors. Of these, systems 1, 2, 3, 5, 7, 8, 10 and 12 belong to one group in the sense that in these systems all the factors considered have contributed equally, if they are present. On the other hand, in systems 4,6,9 and 11 cnet factor has been assigned a weightage of 50 to increase the importance of this factor. The rest of the two systems, system 13 and 14 assign weight to the terms according to generative values.

System1 is the simple system with query expansion model. We have retrieved the relevant document from the collection using improved IR. To create the model, terms in the relevant documents are assigned weight according to the equation 5. In system2 (tf+idf), idf is included for sense of moral, with term frequency (tf). Here tf and idf are considered without any weighting factor. System 3 (tf+Cnet) includes Cnet for network cohesion with term frequency (tf). Here tf and Cnet are considered without any weighting factor. In System 5 (tf+df), document frequency (df) for goal consensus is added with term frequency (tf). In System 7 (tf+idf+Cnet), apart from network cohesion cnet, we have idf, which is included for sense of moral. System 8 (tf+idf+df) considers tf, idf and df, to verify the effect of inclusion of idf with tf and df. System 10 (tf+df+Cnet) considers tf, df and cnet, to verify the effect of inclusion of cnet with tf and df. In System 12 (tf+idf+df+Cnet) considers tf, df, idf and cnet, to verify the effect of inclusion of cnet and idf with tf and df.

Systems 4,6,9 and 11 show the effect of weighted Cnet. As Cnet value is very small compared to the term frequency, we increase this value by multiplying a constant c . Here we have empirically fixed the value of c to be 50. Systems 3,7,10 and 12 consider Cnet without weight value.

In System 13 (tf*idf) assigns weight according to generative value of tf and idf. This system is constructed to evaluate the generative effect of tf and idf. In System 14 (tf*idf*df) assigns weight according to generative value of tf, idf and df.

Table 2 shows the F1-measure, Accuracy and Cost of the various systems discussed above. Fig. 2, 3 and 4 show comparison of the various systems with respect to F1-measure, Accuracy and Cost.

Table 2. F1-Measure of Various Link Detection Systems

Systems	Cost	Improv- ement	F1-Measure	Improv- ement	Accuracy	Improv -ement
ltdf	0.116747	-0.02	0.751696	2.42%	0.889359	1.57%
ltdfnw	0.116747	-0.02	0.751696	2.42%	0.889359	1.57%
tfdf50nw	0.117447	-0.02	0.740638	1.31%	0.886941	1.33%
tfidfd	0.124241	-0.013	0.737838	1.03%	0.882709	0.91%
ltdfd50nw	0.130504	-0.007	0.736842	0.93%	0.879081	0.54%
ltdfd	0.136056	-0.001	0.73385	0.63%	0.875453	0.18%
ltdfdfnw	0.136056	-0.001	0.73385	0.63%	0.875453	0.18%
ltd50nw	0.131328	-0.006	0.732804	0.53%	0.877872	0.42%
ltdfnw	0.136406	-8E-04	0.72846	0.09%	0.874244	0.06%
tf	0.137173	4E-07	0.72751	0.00%	0.87364	0.00%
tfidf	0.141721	0.0045	0.724936	-0.26%	0.870617	-0.30%
ltdfd50nw	0.12098	-0.016	0.724187	-0.33%	0.882104	0.85%
ltdfdfnw	0.122278	-0.015	0.721358	-0.62%	0.880895	0.73%
ltdidf	0.123046	-0.014	0.720339	-0.72%	0.88029	0.67%

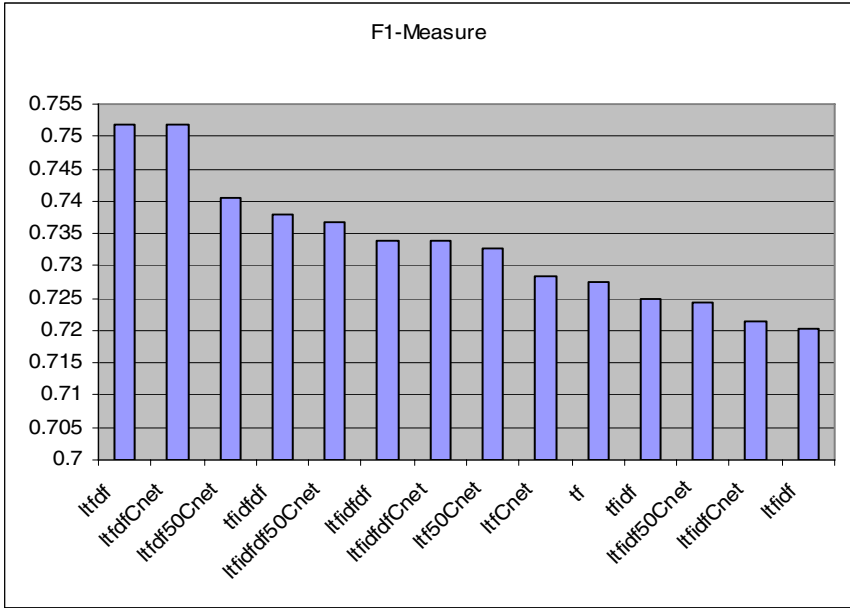


Fig. 2. F1-Measure of various Link Detection Systems

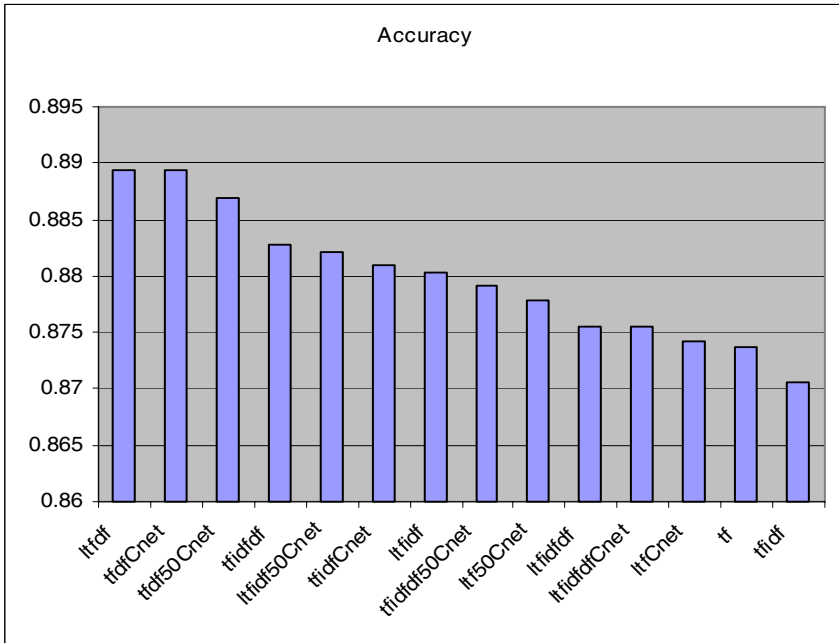


Fig. 3. Accuracy of various Link Detection Systems

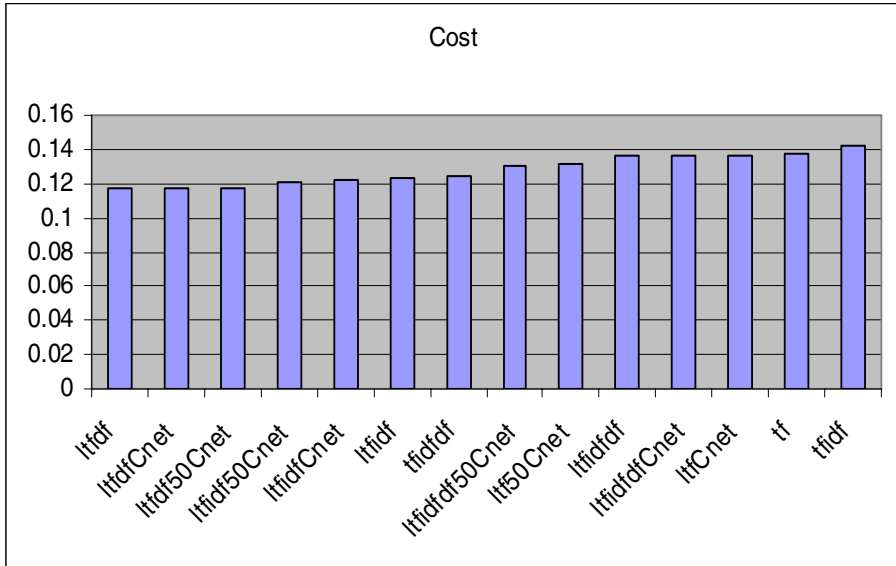


Fig. 4. Cost of various Link Detection Systems

As is evident from Fig. 2-4, System5 is a linear combination of tf-df and shows the best performance compared to all the other systems. System 5 shows 6% increase in F1-measure, 3% increase in accuracy and 0.03724 reductions in cost, when compared to the base system that uses cosine similarity for comparing the two documents without any query expansion technique. This system is able to achieve good true positive at the same time maintaining less false positive and this leads to excellent system performance. System 9 (Tf+df+Cnet), shows same performance as system 5. As the Cnet value is very less, it didn't contribute to the performance of the system.

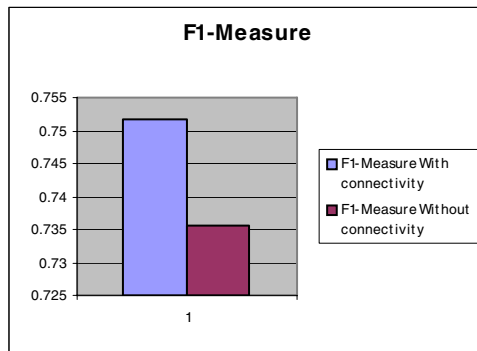


Fig. 5. F1-Measure comparison between model with and without improved IR

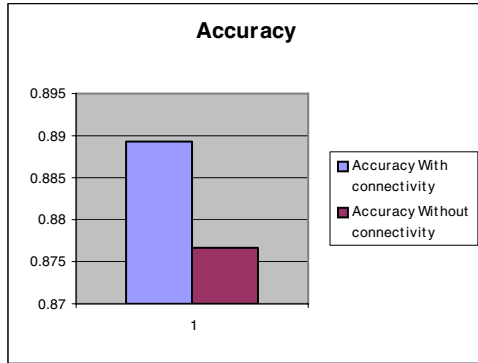


Fig. 6. Accuracy comparison between model with and without improved IR

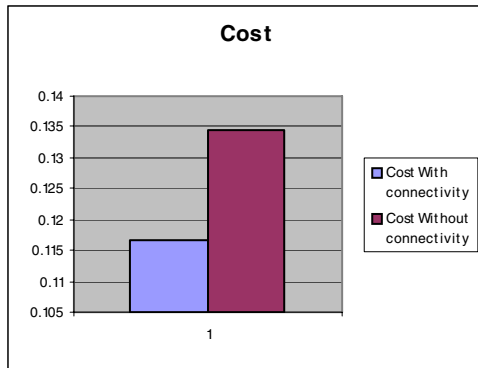


Fig. 7. Cost comparison between model with and without improved IR

System 10 with raised Cnet value, shows the second best performance. It reduces both the true positives and false positives that leads to less F1-measure when compared to System 5. Generative model of $tf \cdot idf \cdot df$ (System 14) increases both the true positives and false positives that causes a reduction in performance when compared to system 9 and 10. Systems 11, 8, 12, 3 and 4 show better performance than the base system that uses only tf .

Performance of generative values are much less comparing to most of the given link detection system. Multiplying tf with idf reduces the performance of the system.

Next point of discussion is how far the improved IR increases the performance of the cohesion model. Fig. 5, 6 and 7 show the comparison between cohesion model with and without improved IR. Experimental results show that there is 1.5% improvement in F1-measure, 1.2% improvement in accuracy and the cost has reduced from 0.135 to 0.115 by using improved information retrieval system.

5 Conclusions and Enhancements

Systems with query expansion technique use improved retrieval system for retrieving relevant documents by using the intra cluster similarity. As the relevant documents are the basis for preparing model for the given input documents, it is important to select high quality relevant documents and reduce the number of irrelevant documents. Social cohesion is used as the basis for assigning weight in Cohesion-Model. We have taken four factors of social cohesion and constructed number of link detection system with various combinations of the basic factors sense of belonging (tf), sense of moral, goal consensus (df) and network cohesion (cnet). Most of the models constructed with various combination of social cohesion factor performed better than base cosine similarity method.

Best performance is achieved by linear combination of tf-df. It shows 6% increase in F1-measure, 3% increase in accuracy and 0.034 reductions in cost, when compared with the base system that use cosine similarity to compare the document pairs. Inclusion of df with tf work better than linear combination of tf-cnet. Inclusion of Cnet with tf, works better than simple tf by reducing the false positives. However the combined effect of tf-df-cnet does not produce the best performance, as its true positives are less. Inclusion of idf increases the false positive, which degrades the system performance compared to system without idf. Experimental results show that the weight given to Cnet affected adversely the performance of link detection system in the current set up. Performance of the link detection system with different weights has to be explored to verify the ultimate effect of Cnet on such systems.

We strongly feel the cohesion model can further be improved by modifying the ways the various factors are measured. As shown in this research work, improvement in IR system will definitely improve the over all link detection.

References

1. Allan, J.: Introduction to Topic Detection and Tracking, Topic Detection and Tracking: Event-based Information Organization, pp. 1–16. Kluwer Academic Publishers, Dordrecht (2002)
2. Allan, J., Carbonell, J., Doddington, G., Yamron, J., Yang, Y.: Topic detection and tracking pilot study: Final report. In: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, pp. 194–218. Morgan Kaufmann publishers, San Francisco (1998)
3. Topic Detection and Tracking (TDT) Project.homepage: <http://www.nist.gov/speech/tests/tdt/>
4. Lavrenko, V.: A Generative Theory of Relevance, PhD Thesis, University Of Massachusetts Amherst (September 2004)
5. Chen, F., Farahat, A., Brants, T.: Multiple Similarity Measures and Source-Pair Information in Story Link Detection. In: Proceedings of HLT-NAACL, pp. 313–320 (2004)
6. Lavrenko, V., Allan, J., DeGuzman, E., LaFlamme, D., Pollard, V., Thomas, S.: Relevance models for topic detection and tracking. In: Proceedings of Human Language Technologies Conference, HLT, pp. 104–110 (2002)

7. Yang, Y., Ault, T., Pierce, T., Lattimer, C.W.: Improving text categorization methods for event tracking. In: SIGIR'00. Proceedings of the 23rd Annual international ACM SIGIR Conference on Research and Development in information Retrieval, Athens, Greece, July 24-28, 2000, pp. 65–72. ACM Press, New York (2000)
8. Farahat, A., Chen, F., Brants, T.: Optimizing Story Link Detection is not Equivalent to Optimizing New Event Detection. In: Dignum, F.P.M. (ed.) ACL 2003. LNCS (LNAI), vol. 2922, pp. 232–239. Springer, Heidelberg (2004)
9. Nallapati, R., Allan, J.: Capturing Term Dependencies using a Language Model based on Sentence Trees. In: CIKM'02, McLean, Virginia (November 4-9, 2002)
10. Lakshmi, K., Mukherjee, S.: An Improved Feature Selection using Maximized Signal to Noise Ratio Technique for TC. In: ITNG 2006. Proceedings of Information Technology: New Generations, pp. 541–546 (April 2006)
11. Allan, J., Lavrenko, V., Frey, D., Khandelwal, V.: UMass at TDT 2000. In: Proceedings of the Topic Detection and Tracking Workshop (2000)
12. Figueroa, M., Lawrence Kincaid, D., Rani, M., Lewis, G. (eds.): Communication for Social Change: An Integrated Model for Measuring the Process and Its Outcomes. The Rockefeller Foundation New York (2002)
13. Raghavan, H., Allan, J.: Using soundex codes for indexing names in ASR documents. In: Proceedings of the HLT NAACL Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval (2004)
14. Lakshmi, K., Mukherjee, S.: Using Cohesion-Model for Story Link Detection System. IJCSNS International Journal of Computer Science and Network Security 7(3), 59–66 (2007)

Improving a State-of-the-Art Named Entity Recognition System Using the World Wide Web

Richárd Farkas¹, György Szarvas^{1,2}, and Róbert Ormándi²

¹ University of Szeged, Department of Informatics
6721 Szeged, Hungary

² Research Group on Artificial Intelligence
of the Hungarian Academy of Sciences and University of Szeged
6721 Szeged, Hungary

rfarkas, szarvas@inf.u-szeged.hu
ormandi.robert@stud.u-szeged.hu

Abstract. The development of highly accurate Named Entity Recognition (NER) systems can be beneficial to a wide range of Human Language Technology applications. In this paper we introduce three heuristics that exploit a variety of knowledge sources (the World Wide Web, Wikipedia and WordNet) and are capable of improving further a state-of-the-art multilingual and domain independent NER system. Moreover we describe our investigations on entity recognition in simulated speech-to-text output. Our web-based heuristics attained a slight improvement over the best results published on a standard NER task, and proved to be particularly effective in the speech-to-text scenario.

Keywords: World Wide Web, web based techniques, named entity recognition, machine learning.

1 Introduction

The identification and classification of Named Entities (NE) in plain text is of key importance in numerous natural language processing applications. In Information Extraction systems NEs generally carry important information about the text itself, and thus are targets for extraction. In machine translation, Named Entities and other sorts of words have to be handled in a different way due to the specific translation rules that apply to them.

We applied the NE Recognition and Classification (NER) system described in [10] which was designed for English language, and also worked with minor changes for Hungarian and domains different from newswire texts (medical records) [11]. To our best knowledge, this system gives the best results on the standard CoNLL-2003 task.

In this paper we investigate three heuristics that utilize online information (the World Wide Web and the Wikipedia online encyclopedia) to improve the performance of this state-of-the-art Named Entity Classification system.

As we plan to integrate our entity recognizer and classifier module into a multi-modal Information Extraction system, we tested the NER system in an artificial

scenario simulating speech-to-text output. As regards the problem of NER on the output of a general purpose speech-to-text application, it assumes that neither capitalization nor punctuation marks are available in the text. These restrictions make entity recognition a more challenging task. Experiments showed that the NER problem can be handled in such circumstances, without a serious loss of classification performance, while our web-based heuristics are particularly useful here.

In this paper we performed experiments for the English newswire NER task only but our heuristics should be portable across languages as long as the appropriate knowledge sources are available for the target language, with sufficient coverage¹.

1.1 Related Work

The NER task was introduced during the nineties as a part of the shared tasks in the Message Understanding Conferences (MUC) [4]. The goal of these conferences was the recognition of proper nouns (*person*, *organization*, *location* names), and other phrases denoting dates, time intervals, and measures in texts from English newspaper articles. The best systems [1] following the MUC task definition achieved outstanding accuracies (nearly 95% F measure).

Later, as a part of the Computational Natural Language Learning (CoNLL) conferences [12], a shared task dealt with the development of systems like this that work for multiple languages and were able to correctly identify *person*, *organization* and *location* names, along with other proper nouns treated as *miscellaneous* entities.

There are some important differences between the CoNLL style task definition and the MUC approach that made NER a much harder problem. The most important is that CoNLL considers only whole phrases classified correctly (which is more suitable for real world applications). The F measure of the best performing systems [7] dropped below 89% for English.

There are several papers in the literature that investigate the usability of online resources for various NE-related tasks. The available systems seek to collect lists of Named Entities belonging to pre-specified classes from the WWW [5][6] or use online information for Named Entity Disambiguation [2], which differ from the problem addressed in this paper. We found no articles on using Web-searches to improve a NER system.

1.2 Structure of the Paper

In the following section we will introduce the NER problem in general, along with the details of the CONLL-2003 English task and the evaluation methodology. We also discuss the learning methods and other main characteristics of the NER system we applied. In section 3 we describe our web-based heuristics designed to improve the classification performance of a state-of-the-art NER system, followed by the description of our experiments on artificial speech-to-text data (Section 4). Experimental results are summarized in the last section along with some concluding remarks.

¹ German, the language with the second largest Wiki encyclopedia has one third entries compared to English.

2 Description of the NER System Applied

In this section we introduce the domain- and language independent NER system we used for our experiments. An NER system in English was trained and tested on a sub-corpus of the Reuters Corpus² (the CoNLL 2003 shared task database), consisting of newswire articles from 1996 provided by Reuters Inc. The data is available free of charge for research purposes and contains texts from diverse domains ranging from sports news to politics and the economy. The best result published in the CoNLL 2003 conference was an F measure of 88.76% obtained from the best individual model [7].

2.1 Evaluation Methodology

To make our results easier to compare with those given in the literature, we employed the same evaluation script that was used during the CoNLL conference shared tasks for entity recognition. This script calculates Precision, Recall and $F_{\beta=1}$ ³ value scores by analyzing the text at the phrase level. This way evaluation is very strict as it can penalize single mistakes in longer entity phrases doubly.

It is worth mentioning that this kind of evaluation places a burden on the learning algorithms as they usually optimize their models based on a different accuracy measure. Fitting this evaluation into the learning phase is not straightforward because of some undesired properties of the formula that can adversely affect the optimization process.

2.2 Complex NER Model

The NER system we use here treats the NER problem as the classification of separate tokens. Following Szarvas et al. [10], we apply decision tree classifiers (with boosting). This way our model is fast to train and evaluate, and incorporates a very

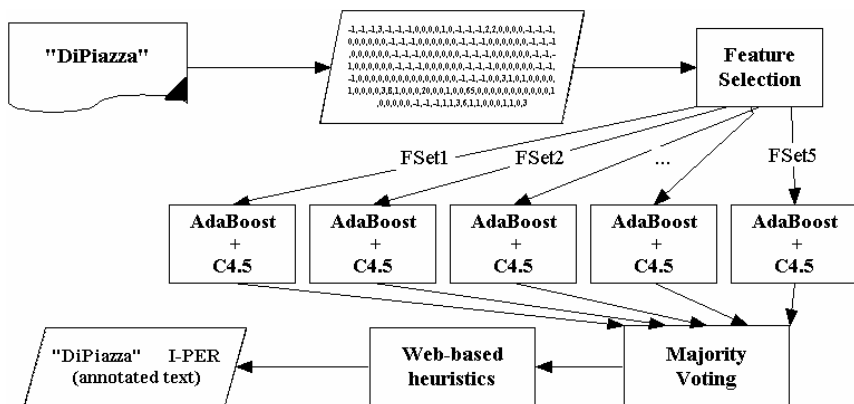


Fig. 1. The structure of our NER system

² <http://www.reuters.com/researchandstandards/>

³ In this paper we always mean $F_{\beta=1}$ under F measure.

rich feature set (described in detail in [10]). The model also takes into account the relationship between consecutive words as well through a window with appropriate window size. The rich feature set enables to split the set, build models on each subset and then recombine their results. Figure 1 sketches the structure of the complex model.

2.3 Feature Set

Initial features. We employed a very rich feature set for our word-level classification model, describing the characteristics of the word itself along with its actual context (a moving window of size four). Our features fell into the following major categories:

- *gazetteers of unambiguous NEs* from the train data: we used the NE phrases which occur more than five in the train texts and got the same label more than 90 percent of the cases,
- *dictionaries* of first names, company types, sport teams, denominators of locations (mountains, city) and so on: we collected 12 English specific lists from the Internet,
- *orthographical features*: capitalization, word length, common bit information about the word form (contains a digit or not, has uppercase character inside the word, regular expressions and so on). We collected the most characteristic character level bi/trigrams from the train texts assigned to each NE class,
- *frequency information*: frequency of the token, the ratio of the token's capitalized and lowercase occurrences, the ratio of capitalized and sentence beginning frequencies of the token,
- *phrasal information*: chunk codes and forecasted class of few preceding words (we used online evaluation),
- *contextual information*: POS codes, sentence position, document zone (title or body), topic code, trigger words (the most frequent and unambiguous tokens in a window around the NEs) from the train text, is the word between quotes and so on.

In our experiments we used a similar feature set splitting strategy as described in [10] to obtain 5 different (but not necessarily disjunctive) sets of features from the categories described above. We used these five sets for bagging similar classifiers to obtain better results than in case of using all features together. The 5 similar AdaBoost+C4.5 boxes and Majority Voting illustrates this in Figure 1.

2.4 Classifiers and Combination Strategies

Boosting [9] and C4.5 [8] are well known algorithms for those who are acquainted with pattern recognition. Boosting has been applied successfully to improve the performance of decision trees in several NLP tasks. A system that made use of AdaBoost and fixed depth decision trees [3] came first on the CoNLL-2002 conference shared task for Dutch and Spanish, but gave somewhat worse results for English and German (it was ranked fifth, and had an F measure of 85.0% for English) in 2003.

As the results of [10] show, the combination of AdaBoost and C4.5 can bring some improvement in classification accuracy and preserves the superiority of decision tree learning in term of CPU time used for training and evaluating a model. In our experiments we used the implementations available in the WEKA [13] library, an open-source data mining software written in Java.

Combination of classifiers. There are several well known meta-learning algorithms in the literature that can lead to a ‘better’ model (in terms of classification accuracy) than those serving as a basis for it, or can significantly decrease the CPU time of the learning phase without loss of accuracy. The decision function used to integrate the five hypotheses (learnt on different subsets of features) was the following: *if any three of the five learners’ outputs coincided we accepted it as a joint prediction, with a forecasted ‘O’ label referring to a non-named entity class otherwise.* This cautious voting scheme is beneficial to system performance as a high rate of disagreement often means a poor prediction rate. For a CoNLL type evaluation it is better to make such mistakes that classifies an NE as non-named entity than to place an NE in a wrong entity class (the latter detrimentally affects precision and recall, while the former only affects the recall of the system).

Here we used the same voting strategy for the baseline system, and tested other alternative voting schemes that exploit online information to assign NE labels in case of disagreement of the learnt models. This will be discussed in detail in the next section.

3 WWW Based Improvement of the NER System

Using online knowledge sources in Human Language Technology (HLT) and Data Mining problems has been an emerging field of research in the past few years. This trend is boosted by several special and interesting characteristics of the World Wide Web. First of all, it provides a practically limitless source of (unlabeled) data to exploit, and, more important it can bring some dynamism to applications. As online data changes and rapidly expands with time, a system can remain up-to-date and extending its knowledge without the need of fine tuning, or any human intervention (like retraining on up-to-date data for example). These features make the Web a very useful source of knowledge for HLT applications as well. On the other hand, the usage of WWW is a new challenge to overcome for language processing applications as data cannot be accessed directly (only via a search engine) and might prove to be time consuming as task-specific pre-processing and collection of data is not feasible.

3.1 Fine-Tuning Phrase Boundaries

A significant part of system errors in NER taggers is caused by the erroneous identification of the beginning (or end) of a longer phrase. Token-level classifiers (like the one we applied here) are especially prone to this as they classify each token of a phrase separately.

We considered a tagged entity as a candidate long-phrase NE if it was followed or preceded by a non-tagged uppercase word, or one/two stop words and an uppercase word. The underlying hypothesis of this heuristic is that if the boundaries were

marked correctly and the surrounding words are not part of the entity, then the number of web-search results for the longer query should be significantly lower (the NE is followed by the particular word in just certain contexts). But in the case of a dislocated phrase boundary, the number of search results for the extended form must be comparable to the results for the shorter phrase (over 0.1%⁴ of it). This means that every time when we found a tagged phrase that received more than 0.1% web query hits in an extended form, we extended the phrase with its neighboring word (or words). This decision function was fine tuned and found to be optimal on the training and development sets of the CoNLL task; the following evaluations have been performed on the CoNLL evaluation set.

This web-based post-processing heuristic improved the performance of the applied NER model from 89.02% to 89.15% F measure. The relatively small improvement is due to the classification error of some extended phrases (this heuristic extended the phrase boundaries precisely in several cases where the class label was assigned incorrectly by the classifier, and those left the system performance unchanged).

3.2 Using the Most Frequent Role in Uncertain Cases

Some examples are easier to classify for a given model than others. In our applied NER system, the final decision was obtained by applying the majority voting procedure of 5 classifiers (which were all trained on different sets of features). A simple way of interpreting the uncertainty of a decision is to measure the level of disagreement among the individual models. We considered a token as a difficult or uncertain example if no more than 2 models gave coinciding decisions (we should mention here that each models chose the most probable of 5 different possible answers, so this indeed meant a high level of uncertainty).

Our hypothesis here was that the most frequent role of a named entity can be statistically useful information. Thus we did the following: if the system was unable to decide the class label of a phrase (it could not find evidence in the context of the certain phrase) then we mined the most frequent usage of the corresponding NE using the WWW and took that as prediction.

The most frequent role searching method we applied here was inspired by the category extraction methods of Etzioni et al. [6]. This approach works by invoking several special Google queries in order to find such noun phrases following or preceding the pattern that is a category name for a particular class. The following queries were used to obtain category names from web search results:

NP such as NE
NP including NE
NP especially NE
NE is a NP
NE is the NP
NE and other NP
NE or other NP

⁴ We sought to keep the evaluation set blind. All the heuristics were fine-tuned on CoNLL-2003 development set.

Category names from the training data. We used the lists of unambiguous NEs collected from the training data to acquire common NE category names. We sent Google queries for NEs in the training data and all the patterns shown above. The heads of the corresponding NPs were extracted from the snippets of the best ten Google responses.

We found 173 reliable category names by performing a limited number of Google queries. Using these category lists as a disambiguator (we assigned the class sharing the most words in common with those extracted for the given NE) when the NER system was unable to give a reliable prediction was beneficial to overall system performance. The system F Measure improved from 89.15% to 89.28%. We should mention here that the baseline NER system labeled these examples as non-entities, whose prediction was incorrect in the majority of the cases.

Enriching category lists using WordNet. We enlisted the help of a linguist expert to determine the WordNet synset corresponding to each category name we found and give its most common substituting synset (the one highest in hypo/hypernym hierarchy) that was still usable as a category name for the particular NE class. Using these WordNet synsets we extended our category lists (to a size of 19537) with all literals that appear in their hyponym subtree (with sense #1). This additional knowledge further improved the F measure of the NER system to 89.35%.

4 Experiments on Speech-to-Text Data

Named Entity Recognition on the output of a speech-to-text system has to handle the problem of several missing features (like capitalization) that are particularly useful for entity recognition.

We used the same data as for the experiments described above, but modified the text so it looked as if it had been obtained from a speech-to-text system. First we converted all tokens to lowercase, thus the feature that is undoubtedly the most important for NER became unavailable. Second, we removed all punctuation marks from the original corpus (they do not appear explicitly in the audio stream, only in the accent hence it is doubtful that any punctuation can be retrieved efficiently). This means we assumed that all word forms were recognized correctly.

In the majority of cases, consecutive Named Entities either follow each other with a separating punctuation mark (enumerations), or belong to different classes. In the first case, a non-labeled token separates the two phrases, while in the second case the different class labels identify the boundaries. Rarely do two or more NEs of the same type appear consecutively in a sentence. In such cases the phrasal boundaries must be marked with a tag ('B-' instead of the common 'I-' prefix). We changed 'I-' tags to 'B-' where it was necessary in the simulated speech-to-text data to retain the correct phrase boundaries. This conversion resulted in over ten times more consecutive NEs

(those separated with ‘B-‘ tag), and hence the separation of such phrases became no longer negligible.⁵

We should add here that this simulation of the output of a speech-to-text system seemed obvious for two reasons. First, we wanted to test how a NER system behaves in significantly different circumstances, not a speech-to-text system itself. Second, by doing this we could avoid the need for a NE-labeled real speech database and also have better grounds for comparison between written text and speech-to-text output as we used a standard database. The performance of the baseline NER system on this converted text decreased to 81.1% $F_{\beta=1}$. Even though this simplification does not take into account that real speech-to-text data would certainly contain word errors, it fits to our purposes well (it is capable of demonstrating the usability of online knowledge sources to improve NER in speech-to-text data).

4.1 Identifying Consecutive NEs

As we stated above ‘B-‘ tags are even more common in texts obtained from a speech-to-text system due to the absence of punctuation marks. We exploited the encyclopedic knowledge of Wikipedia to enable our system to distinguish between long phrases and consecutive entities.

The B-tag heuristic. We queried the Wikipedia site for all entities that had two or more tokens. If we found an article sharing the same title as the whole query, or the majority of the occurrences of the phrase in the Google snippets occurred without punctuation marks inside, we treated the query phrase as a single entity. If a punctuation mark was inside the phrase in the majority of the cases, we separated the phrase at the position of the punctuation mark. This method allowed us to separate phrases like ‘Golan Heights | Israel’. If there was no hit for the query in the Wikipedia, but we were able to find a specific article for two or more parts of the query, we put phrase boundaries following the Wiki entries. This way we identified successfully phrases like ‘Taleban | MiG-19’ and many enumerations that lacked the separating commas due to the removal of punctuation marks from the data. We made use of a first names list here containing 3217 first names which allowed us to avoid the erroneous separation of full names (First name, Last name pairs). Of course a more comprehensive first names list would be beneficial. Our system suffered from the lack of Romanian or Arabic First names. This heuristic improved the overall performance of the NER tagger on speech-to-text data by a significant 1.42% (8,1% error reduction). The heuristic itself managed to recognize the ‘B-‘ tags with an $F_{\beta=1}$ -measure of 75.19% (precision 71.7%; recall 79.03%).

We should also mention here that some of the ‘B-‘ phrases in the CoNLL database are arguably consecutive NEs, but are actually single entities (e.g. ‘English Moslems’ or ‘City State’ phrases like ‘Rochester NY’). Our heuristic does not divide up such cases as they usually seem to be single NEs for the online encyclopedia – and they

⁵ Most of the best performing NER systems deliberately ignore the separation of consecutive phrases as they are too sparse to handle efficiently in written text data. This problem has no significant effect on performance either (there are only 20 ‘B-‘ tokens out of 50,000 in the CoNLL-2003 test dataset).

can be treated as single entities as well in an Information Extraction system. Without these cases the recall of our system would have been even higher.

5 Summary of the Results

A brief summary of the heuristic improvements achieved on the various systems can be seen in Table 1. Here we show the system described in Section 2 (and in [10] in more details), *Base NER*; its voting with the 2 best CoNLL systems, *Voting*; the system described in Section 2 on the speech-to-text data, *Speech-to-text*. The boundary heuristic is not applicable in the speech-to-text task, because it is dependant on the capitalization of the context. The *Voting* column adds a further voting level to the system (not showed in Figure 1.), it is obtained by the majority voting the best performing CoNLL systems and *Base NER*. This hybrid method was also discussed in [10]; we show here that the improvement of our web based heuristics carries over to this hybrid model also.

Table 1. Results of the three heuristics, $F_{\beta=1}$

	Base NER	Voting	Speech-to-text
Baseline	89.02%	91.40%	81.10%
B-tag	89.02%	91.40%	82.52%
Boundary id.	89.15%	91.51%	n/a
Most freq. role	89.35%	91.67%	82.64%

6 Conclusion

The aim of this paper was to show the potentials of the WWW in HLT problems like named entity recognition. Our heuristics are based on the assumption that, even though the World Wide Web contains much useless and incorrect information, regarding our simple features the correct usage of language dominates over misspellings and other sorts of noise. Our experiments confirmed this hypothesis. We showed experimentally that these heuristics could further improve a state-of-the-art NER system on the standard text processing task and they proved to be particularly useful on a more challenging simulated speech-to-text task. We believe that our results are valuable due to two main reasons: first, we managed to give improvements on a top performing model for the task of NER that is of great importance even if the improvement is slight. Second, we showed that the WWW can be exploited with significant success to overcome the drawback caused by the lack of certain information that is extremely important and characteristic for certain HLT applications (like the absence of punctuation or capitalization in NER).

Acknowledgments. We would like to thank the anonymous reviewers for their valuable comments.

References

1. Bikel, D.M., Schwartz, R.L., Weischedel, R.M.: An algorithm that learns what's in a name. *Machine Learning* 34(1-3), 211–231 (1999)
2. Bunescu, R., Paşca, M.: Using Encyclopedic Knowledge for Named Entity Disambiguation. In: *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics* (2006)
3. Carreras, X., Márques, L., Padró, L.: Named Entity Extraction using AdaBoost. *Proceedings of CoNLL-2002, Taipei, Taiwan*, pp. 167–170 (2002)
4. Chinchor, N.: MUC-7 Named Entity Task Definition. In: *Proceedings of Seventh MUC* (1998)
5. Cimiano, P., Handschuh, S., Staab, S.: Towards the self-annotating web. In: *Proceedings of the 13th WWW Conference* (2004)
6. Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Unsupervised named-entity extraction from the web: an experimental study. *Artificial Intelligence* 165(1), 91–134 (2005)
7. Florian, R., Ittycheriah, A., Jing, H., Zhang, T.: Named Entity Recognition through Classifier Combination. In: *Proceedings of CoNLL-2003* (2003)
8. Quinlan, R.: *C4.5: Programs for machine learning*. Morgan Kaufmann, San Francisco (1993)
9. Shapire, R.E.: The Strength of Weak Learnability. *Machine Learnings* 5, 197–227 (1990)
10. Szarvas, Gy., Farkas, R., Kocsor, A.: A multilingual named entity recognition system using boosting and c4.5 decision tree learning algorithms. In: Todorovski, L., Lavrač, N., Jantke, K.P. (eds.) *DS 2006. LNCS (LNAI)*, vol. 4265, pp. 267–278. Springer, Heidelberg (2006)
11. Szarvas, G., Farkas, R., Iván, S., Kocsor, A., Busa-Fekete, R.: An iterative method for the de-identification of structured medical text. *Workshop on Challenges in Natural Language Processing for Clinical Data* (2006)
12. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: *Proceedings of CoNLL-2003* (2003)
13. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, San Francisco (2005)

ISOR-2: A Case-Based Reasoning System to Explain Exceptional Dialysis Patients

Olga Vorobieva¹, Alexander Rumyantsev², and Rainer Schmidt¹

¹ Institute for Medical Informatics and Biometry, University of Rostock, Germany
rainer.schmidt@medizin.uni-rostock.de

² Pavlov State Medical University, St.Petersburg, Russia

Abstract. In medicine many exceptions occur. In medical practice and in knowledge-based systems too, it is necessary to consider them and to deal with them appropriately. In medical studies and in research, exceptions shall be explained. We present a system that helps to explain cases that do not fit into a theoretical hypothesis. Our starting points are situations where neither a well-developed theory nor reliable knowledge nor a priori a proper case base is available. So, instead of reliable theoretical knowledge and intelligent experience, we have just some theoretical hypothesis and a set of measurements.

In this paper, we propose to combine CBR with a statistical model. We use CBR to explain those cases that do not fit the model. The case base has to be set up incrementally, it contains the exceptional cases, and their explanations are the solutions, which can be used to help to explain further exceptional cases.

1 Introduction

In medicine many exceptions occur. In medical practice and in knowledge-based systems too, these exceptions have to be considered and have to be dealt with appropriately. In ISOR-1, we demonstrated advantages of case-based reasoning (CBR) in situations where a theoretically approved medical decision does not produce the desired and usually expected results [1, 2].

In medical studies and in research, exceptions shall be explained. The present research is a logical continuation of our previous work. It is still the same system and the same structure of dialogues, but now ISOR-2 deals with situations where neither a well-developed theory nor reliable knowledge nor a proper case base is available. So, instead of reliable theoretical knowledge and intelligent experience, we now have just some theoretical hypothesis and a set of measurements. In such situations the usual question is, how do measured data fit to theoretical hypotheses. To statistically confirm a hypothesis it is necessary, that the majority of cases fit the hypothesis. Mathematical statistics determines the exact quantity of necessary confirmation [3]. However, usually a few cases do not satisfy the hypothesis. We examine these cases to find out why they do not satisfy the hypothesis. ISOR-2 offers a dialogue to guide the search for possible reasons in all components of the data system. The exceptional cases belong to the case base. This approach is justified by a certain mistrust of

statistical models by doctors, because modelling results are usually unspecific and “average oriented” [4], which means a lack of attention to individual “imperceptible” features of concrete patients.

The usual CBR assumption is that a case-base with complete solutions is available. Our approach starts in a situation where such a case-base is not available but has to be set up incrementally (figure 1). So, we must

1. Construct a model,
2. Point out the exceptions,
3. Find causes why the exceptional cases do not fit the model, and
4. Develop a case-base.

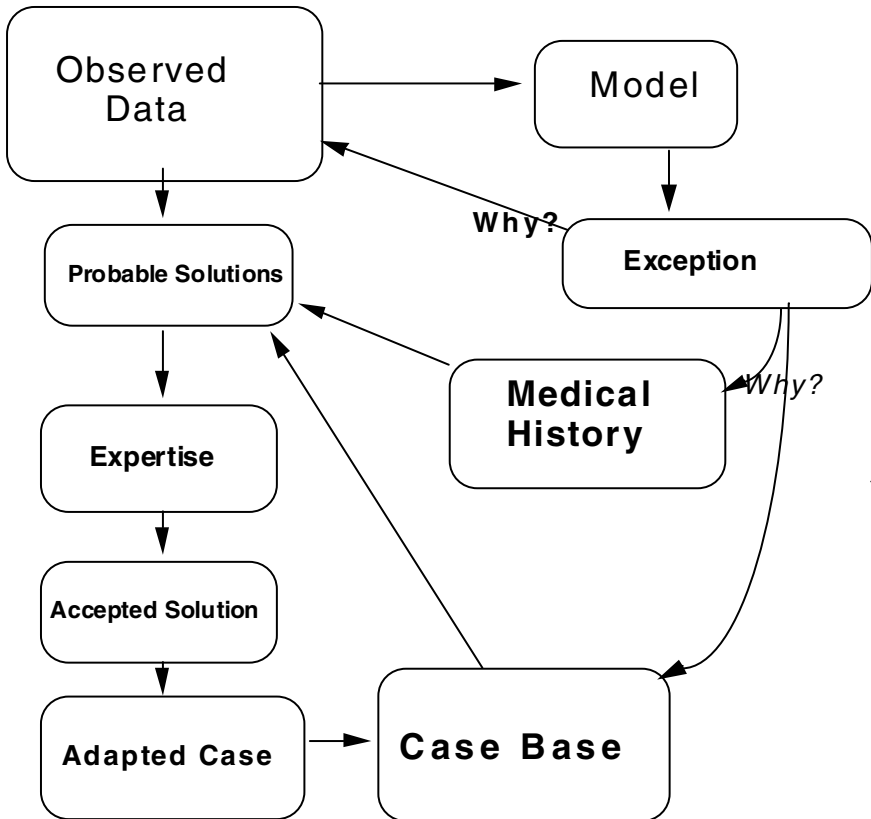


Fig. 1. ISOR-2's general program flow

So, we combine case-based reasoning (CBR) with a model, in this specific situation with a statistical one. The idea to combine CBR with other methods is not new. For example Care-Partner resorts to a multi-modal reasoning framework for the co-operation of CBR and Rule-based Reasoning (RBR) [5]. Another way of combining hybrid rule bases with CBR is discussed by Prentzas and Hatzilgeroudis

[6]. The combination of CBR and model-based reasoning is discussed in [7]. Statistical methods are used within CBR mainly for retrieval and retention (e.g. [8, 9]). Arshadi proposes a method that combines CBR with statistical methods like clustering and logistic regression [10].

1.1 Dialysis and Fitness

Hemodialysis means stress for a patient's organism and has significant adverse effects. Fitness is the most available and a relative cheap way of support. It is meant to improve a physiological condition of a patient and to compensate negative dialysis effects. One of the intended goals of this research is to convince the patients of the positive effects of fitness and to encourage them to make efforts and to go in for sports actively. This is important because dialysis patients usually feel sick, they are physically weak, and they do not want any additional physical load [11].

At our University clinic in St. Petersburg, a specially developed complex of physiotherapy exercises including simulators, walking, swimming etc. was offered to all dialysis patients but only some of them actively participated, whereas some others participated but were not really active. The purpose of this fitness offer was to improve the physical conditions of the patients and to increase the quality of their lives.

2 Incremental Development of an Explanation Model for Exceptional Dialysis Patients

For each patient a set of physiological parameters is measured. These parameters contain information about burned calories, maximal power achieved by the patient, his oxygen uptake, his oxygen pulse (volume of oxygen consumption per heart beat), lung ventilation and others. There are also biochemical parameters like haemoglobin and other laboratory measurements. More than 100 parameters were planned for every patient. But not all of them were really measured.

Parameters are supposed to be measured four times during the first year of participating in the fitness program. There is an initial measurement followed by a next one after three months, then after six months and finally after a year. Unfortunately, since some measurements did not happen, many data are missing. Therefore the records of the patients often contain different sets of measured parameters.

It is necessary to note that parameter values of dialysis patients essentially differ from those of non-dialysis patients, especially of healthy people, because dialysis interferes with the natural, physiological processes in an organism. In fact, for dialysis patients all physiological processes behave abnormally. Therefore, the correlation between parameters differs too.

For statistics, this means difficulties in applying statistical methods based on correlation and it limits the usage of a knowledge base developed for normal people. Non-homogeneity of observed data, many missing values, many parameters for a relatively small sample size, all this makes our data set practically impossible for usual statistical analysis.

Our data set is incomplete therefore we must find additional or substitutional information in other available data sources. They are data bases – the already existent Individual Base and the sequentially created Case Base and the medical expert as a special source of information.

2.1 Setting Up a Model

We start with a medical problem that has to be solved based on given data. In our example it is: "Does special fitness improve the physiological condition of dialysis patients?" More formally, we have to compare physical conditions of active and non-active patients. Patients are divided into two groups, depending on their activity, active patients and non-active ones.

According to our assumption, active patients should feel better after some months of fitness, whereas non-active ones should feel rather worse. We have to define the meaning of "feeling better" and "feeling worse" in our context. A medical expert selects appropriate factors from ISOR's menu. It contains the list of field names from the observed data base.

The expert selects the following main factors

- F1: O2PT - Oxygen pulse by training
- F2: MUO2T - Maximal Uptake of Oxygen by training
- F3: WorkJ – performed Work (Joules) during control training

Subsequently the "research time period" has to be determined. Initially, this period was planned to be twelve months, but after a while the patients tend to give up the fitness program. This means, the longer the time period, the more data are missing. Therefore, we had to make a compromise between time period and sample size. A period of six months was chosen.

The next question is whether the model shall be quantitative or qualitative? The observed data are mostly quantitative measurements. The selected factors are of quantitative nature too. On the other side, the goal of our research is to find out whether physical training improves or worsens the physical condition of the dialysis patients.

We do not have to compare one patient with another patient. Instead, we compare every patient with his own situation some months ago, namely just before the start of the fitness program. The success shall not be measured in absolute values, because the health statuses of patients are very different. Thus, even a modest improvement for one patient may be as important as a great improvement of another. Therefore, we simply classify the development in two categories: "better" and "worse". Since the usual tendency for dialysis patients is to worsen in time, we added those few patients where no changes could be observed to the category "better".

The three main factors are supposed to describe the changes of the physical conditions of the patients. The changes are assessed depending on the number of improved factors:

- Weak version of the model: at least one factor has improved
- Medium version of the model: at least two factors have improved
- Strong version of the model: all three factors have improved

The final step means to define the type of model. Popular statistical programs offer a large variety of statistical models. Some of them deal with categorical data. The easiest model is a 2x2 frequency table. Our “Better/ Worse” concept fits this simple model very well. So the 2x2 frequency table is accepted. The results are presented in table 1.

Table 1. Results of Fisher’s Exact Test, performed with an interactive Web-program: <http://www.matforsk.no/lola/fisher.htm>

Improvement mode	Patient’s physical condition	Active	Non-active	Fisher Exact p
Strong	Better	28	2	< 0.0001
	Worse	22	21	
Medium	Better	40	10	< 0.005
	Worse	10	12	
Weak	Better	47	16	< 0.02
	Worse	3	6	

According to our assumption after six months of active fitness the conditions of the patients should be better.

Statistical analysis shows a significant dependence between the patient’s activity and improvement of their physical condition. Unfortunately, the most popular Pearson Chi-square test is not applicable here because of the small values “2” and “3” in table 1. But Fisher’s exact test [3] can be used. In the three versions shown in table 1 a very strong significance can be observed. The smaller the value of p is, the more significant the dependency.

Exceptions. So, the performed Fisher test confirms the hypothesis that patients doing active fitness achieve better physical conditions than non-active ones. However, there are exceptions, namely active patients whose health conditions did not improve.

Exceptions should be explained. Explained exceptions build the case base. According to table 1, the stronger the model, the more exceptions can be observed and have to be explained. Every exception is associated with at least two problems. The first one is “Why did the patient’s condition get worse?” Of course, “worse” is meant in terms of the chosen model. Since there may be some factors that are not included in the model but have changed positively, the second problem is “What has improved in the patient’s condition?” To solve this problem we look for significant factors where the values improved.

In the following section we explain the set-up of a case base on the strongest model version.

2.2 Setting Up a Case Base

We intend to solve both problems (mentioned above) by means of CBR. So we begin to set up the case-base up sequentially. That means, as soon as an exception is explained, it is incorporated into the case-base and can be used to help explaining

further exceptional cases. We chose a random order for the exceptional cases. In fact, we took them in alphabetical order.

The retrieval of already explained cases is performed by keywords. The main keywords are the usual ISOR ones, namely “problem code”, “diagnosis”, and “therapy”. In the situation of explaining exceptions for dialysis patients the instantiations of these keywords are “adverse effects of dialysis” (diagnosis), “fitness” (therapy), and two specific problem codes. Besides the main ISOR keywords additional problem specific ones are used. Here the additional key is the number of worsened factors. Further keywords are optional. They are just used when the case-base becomes bigger and retrieval is not simple any longer.

However, ISOR-2 does not only use the case-base as knowledge source but further sources are involved, namely the patient’s individual base (his medical history) and observed data (partly gained by dialogue with medical experts). Since in the domain of kidney disease and dialysis the medical knowledge is very detailed and much investigated but still incomplete, it is unreasonable to attempt to create an adequate knowledge base. Therefore, a medical expert, observed data, and just a few rules serve as medical knowledge sources.

2.2.1 Expert Knowledge and Artificial Cases

Expert’s knowledge can be used in many different ways. First we use it to acquire rules, second it can be used to select appropriate items from the list of retrieved solutions, to propose new solutions and last but not least – to create artificial cases.

Initially artificial cases are created by an expert, afterwards they can be used in the same way as real cases. They are created in the following situation. An expert points out a factor F as a possible solution for a query patient. Since many values are missing, it can happen that just for the query patient values of factor F are missing. The doctor’s knowledge in this case can not be applied, but it is sensible to save it anyway. Principally there are two different ways to do this. The first one means to generate a correspondent rule and to insert it into ISOR-2’s algorithms. Unfortunately, this is very complicated, especially to find an appropriate way for inserting such a rule. The alternative is to create an artificial case. Instead of a patient’s name an artificial case number is generated. The other attributes are either inherited from the query case or declared as missing. The retrieval attributes are inherited. This can be done by a short dialogue (figure2) and ISOR-2’s algorithms remain intact. Artificial cases can be treated in the same way as real cases, they can be revised, deleted, generalised etc.

2.2.2 Solving the Problem “Why Did Some Patients Conditions Became Worse?”

As results we obtain a set of solutions of different origin and different nature. There are three categories of solution: additional factor, model failure, and wrong data.

Additional factor. The most important and most frequent solution is the influence of an additional factor. Only three main factors are obviously not enough to describe all medical cases. Unfortunately, for different patients different additional factors are important. When ISOR-2 has discovered an additional factor as explanation for an exceptional case, the factor has to be confirmed by a medical expert before it can be

accepted as a solution. One of these factors is Parathyroid Hormone (PTH). An increased PTH level sometimes can explain a worsened condition of a patient [11]. PTH is a significant factor, but unfortunately it was measured only for some patients.

Some exceptions can be explained by indirect indications. One of them is a very long time of dialysis (more than 60 months) before a patient began with the training program.

Another solution was a phosphorus blood level. We used the principle of artificial cases to introduce the factor phosphorus as a new solution. One patient's record contained many missing data. The retrieved solution meant high PTH, but PTH data in the current patient's record was missing too. The expert proposed an increased phosphorus level as a possible solution. Since data about phosphorus data was missing too, an artificial case was created, who inherited all retrieval attributes of the query case while the other attributes were recorded as missing. According to the expert high phosphorus can explain the solution. Therefore it is accepted as an artificial solution or a solution of an artificial case.

Model failure. We regard two types of model failures. One of them is deliberately neglected data. Some data had been neglected. As a compromise we just considered data of six months and further data of a patient might be important. In fact, three of the patients did not show an improvement in the considered six month but in the following six months. So, they were wrongly classified and should really belong to the "better" category. The second type of model failure is based on the fact that the two-category model was not precise enough. Some exceptions could be explained by a tiny and not really significant change in one of the main factors. Wrong data are usually due to a technical mistake or to not really proved data. For example, one patient was reported as actively participating in the fitness program but really was not.

2.2.3 Solving the Problem "What in the Patient's Condition Became Better?"

There are at least two criteria to select factors for the model. Firstly, a factor has to be significant, and secondly there must be enough patients for which this factor was measured at least for six months. So, some principally important factors were initially not taken into account because of missing data. The list of solutions includes these factors (figure 2): haemoglobin, maximal power (watt) achieved during control training. Oxygen pulse and oxygen uptake were measured in two different situations, namely during the training under loading and before training in a rest state. Therefore we have two pairs of factors: oxygen pulse in state of relax (O2PR) and during training (O2PT); maximal oxygen uptake in state of relax (MUO2R) and during training (MUO2T). Measurements made in a state of relax are more indicative and significant than those made during training. Unfortunately, most measurements were made during training. Only for some patients correspondent measurements in relax state exist. Therefore O2PT and MUO2T were accepted as main factors and were taken into the model. On the other side, O2PR and MUO2R serve as solutions for the current problem.

In the case base every patient is represented by a set of cases, every case represents a specific problem. This means that a patient is described from different points of view and accordingly different problem keywords are used for retrieval.

2.3 Illustration of ISOR-2's Program Flow

Figure 2 shows the main dialogue of ISOR-2 where the user at first sets up a model (steps one to four), subsequently gets the result and an analysis of the model (steps five to eight), and then attempts to find explanations for the exceptions (steps nine and ten). Finally the case base is updated (steps eleven and twelve). On the menu (figure 2) we have numbered the steps and explain them in detail.

At first the user has to set up a model. To do this he has to select a grouping variable. In this example CODACT was chosen. It stands for “activity code” and means that active and none active patients are to be compared. Provided alternatives are the sex and the beginning of the fitness program (within the first year of dialysis or later). In another menu the user can define further alternatives. Furthermore, the user has to select a model type (alternatives are “strong”, “medium”, and “weak”), the length of time that should be considered (3, 6 or 12 months), and main factors have to be selected. The list contains the factors from the observed database. In the example three factors are chosen: O2PT (oxygen pulse by training), MUO2T (maximal oxygen uptake by training), and WorkJ (work in joules during the test training). In the menu list, the first two factors have alternatives: “R” instead of “T”, where “R” stands for state of rest.

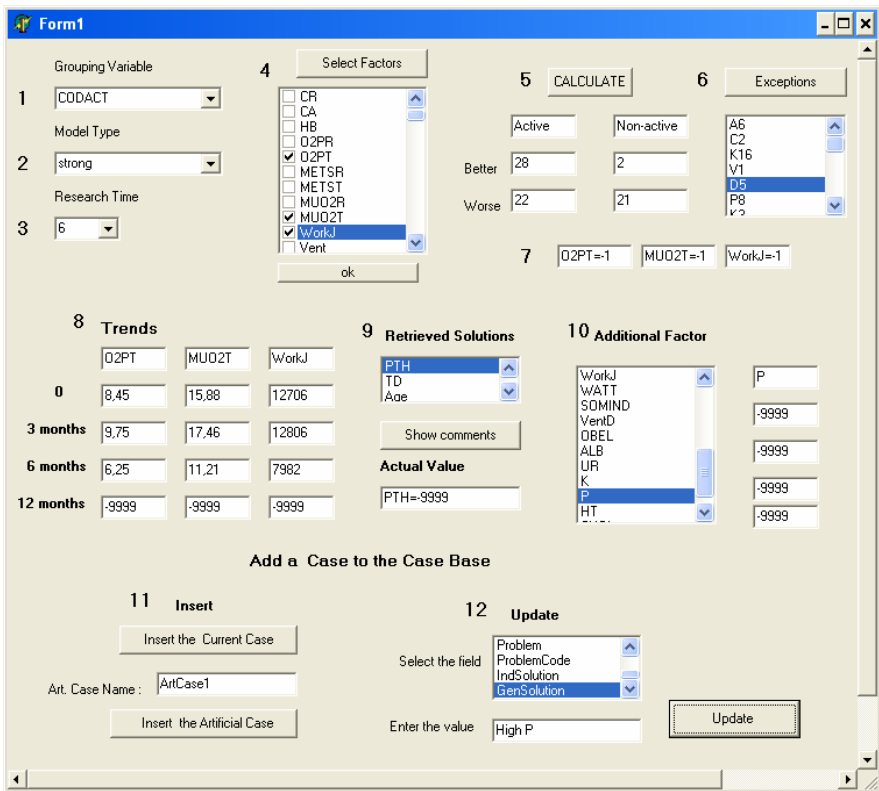


Fig. 2. ISOR-2's main menu

When the user has selected these items, the program calculated the table. “Better” and “worse” are meant in the sense of the chosen model, in the example of the strong model. ISOR-2 does not only calculate the table but additionally extracts the exceptional patients from the observed database. In the menu, the list of exceptions shows the code names of the patients. In the example patient “D5” is selected” and all further data belong to this patient. The goal is to find an explanation for the exceptional case “D5”. In point seven of the menu it is shown that all selected factors worsened (-1), and in point eight the factor values according to different time intervals are depicted. All data for twelve months are missing (-9999).

The next step means creating an explanation for the selected patient “D5”. From the case base ISOR-2 retrieves general solutions. The first retrieved one in this example, the PTH factor, denotes that the increased Parathyroid hormone blood level may explain the failure. Further theoretical information (e.g. normal values) about a selected item can be received by pressing the button “show comments”. The PTH value of patient “D5” is missing (-9999). From menu point ten the expert user can select further probable solutions. In the example an increased phosphorus level (P) is suggested. Unfortunately, phosphorus data are missing too. However, the idea of an increased phosphorus level as a possible solution shall not be lost. So, an artificial case has to be generated.

The final step means inserting new cases into the case base. There are two sorts of cases, query cases and artificial cases. Query cases are stored records of real patients from the observed database. These records contain a lot of data but they are not structured. The problem and its solution transform them into cases and they get a place in the case base.

Artificial cases inherit the key attributes from the query cases (point seven in the menu). Other data may be declared as missing, by the update function data can be inserted. In the example of the menu, the generalised solution “High P” is inherited, it may be retrieved as a possible solution (point 9 of the menu) for future cases.

2.4 Example: A New Problem

Above we described just one of many problems that can arise based on the observed data set and that can be solved and analysed by the dialogue of figure 2. The question to be discussed is “Does it make sense to begin with the fitness program during the first year of dialysis?” The question arises, because the conditions of the patients are considered to be unstable during their first year of dialysis. So, the question is expressed in this way “When shall patients begin with the fitness program, earlier or later?” The term “Earlier” is defined as “during the first year of dialysis”. The term “Later” means that they begin with their program after at least one year of dialysis. To answer this question we consider two groups of active patients, those who began their training within the first year of dialysis and those who began it later (table 2).

Table 2. Changed conditions for active patients

	Earlier	Later
Better	18	10
Worse	6	16

According to Fisher's Exact Test dependence can be observed, with $p < 0,05$. However, it is not as it was initially expected. Since patients are considered as unstable during their first year of dialysis, the assumption was that an earlier beginning might worsen conditions of the patients. But the test revealed that the conditions of active patients who began with their fitness program within the first year of dialysis improved more than those of patients starting later.

However, there are 6 exceptions, namely active patients starting early and their conditions worsened. The explanations of them are high PTH or high phosphorus level.

3 Conclusion

In this paper, we have proposed to use CBR in ISOR-2 to explain cases that do not fit a statistical model. Here we presented one of the simplest statistical models. However, it is relatively effective, because it demonstrates statistically significant dependencies, in our example between fitness activity and health improvement of dialysis patients, where the model covers about two thirds of the patients, whereas the other third can be explained by applying CBR. Since we have chosen qualitative assessments (better or worse), very small changes appear to be the same as very large ones. We intend to define these concepts more precisely, especially to introduce more assessments. The presented method makes use of different sources of knowledge and information, including medical experts. It seems to be a very promising method to deal with a poorly structured database, with many missing data, and with situations where cases contain different sets of attributes.

Acknowledgement

We thank Professor Aleksey Smirnov, director of the Institute for Nephrology of St-Petersburg Medical University and Natalia Korosteleva, researcher at the same Institute for collecting and managing the data.

References

1. Schmidt, R., Vorobieva, O.: Case-Based Reasoning Investigation of Therapy Inefficacy. *Knowledge-Based Systems* 19(5), 333–340 (2006)
2. Schmidt, R., Vorobieva, O.: Adaptation and Medical Case-Based Reasoning Focusing on Endocrine Therapy Support. In: Miksch, S., Hunter, J., Keravnou, E.T. (eds.) *AIME 2005. LNCS (LNAI)*, vol. 3581, pp. 308–317. Springer, Heidelberg (2005)
3. Kendall, M.G., Stuart, A.: *The advanced theory of statistics*, 4th edn. Macmillan publishing, New York (1979)
4. Hai, G.A.: *Logic of diagnostic and decision making in clinical medicine*. Politekhnica publishing, St. Petersburg (2002)
5. Bichindaritz, I., Kansu, E., Sullivan, K.M.: Case-based Reasoning in Care-Partner. In: Smyth, B., Cunningham, P. (eds.) *EWCBR 1998. LNCS (LNAI)*, vol. 1488, pp. 334–345. Springer, Heidelberg (1998)

6. Prentzas, J., Hatzilgeroudis, I.: Integrating Hybrid Rule-Based with Case-Based Reasoning. In: Craw, S., Preece, A.D. (eds.) ECCBR 2002. LNCS (LNAI), vol. 2416, pp. 336–349. Springer, Heidelberg (2002)
7. Shuguang, L., Qing, J., George, C.: Combining case-based and model-based reasoning: a formal specification. In: Proc APSEC'00, p. 416 (2000)
8. Corchado, J.M., Corchado, E.S., Aiken, J., et al.: Maximum likelihood Hebbian learning based retrieval method for CBR systems. In: Ashley, K.D., Bridge, D.G. (eds.) ICCBR 2003. LNCS, vol. 2689, pp. 107–121. Springer, Heidelberg (2003)
9. Rezvani, S., Prasad, G.: A hybrid system with multivariate data validation and Case-based Reasoning for an efficient and realistic product formulation. In: Ashley, K.D., Bridge, D.G. (eds.) ICCBR 2003. LNCS, vol. 2689, pp. 465–478. Springer, Heidelberg (2003)
10. Arshadi, N., Jurisica, I.: Data Mining for Case-based Reasoning in high-dimensional biological domains. *IEEE Transactions on Knowledge and Data Engineering* 17(8), 1127–1137 (2005)
11. Davidson, A.M., Cameron, J.S., Grünfeld, J.-P., et al. (eds.): *Oxford Textbook of Nephrology*, vol. 3. Oxford University Press, Oxford (2005)

The Role of Prototypical Cases in Biomedical Case-Based Reasoning

Isabelle Bichindaritz

University of Washington, 1900 Commerce Street, Box 358426,
Tacoma, WA 98402, USA
ibichind@u.washington.edu

Abstract. Representing biomedical knowledge is an essential task in biomedical informatics intelligent systems. Case-based reasoning (CBR) holds the promise of representing contextual knowledge in a way that was not possible before with traditional knowledge representation and knowledge-based methods. A main issue in biomedical CBR has been dealing with maintenance of the case base, and particularly in medical domains, with the rate of generation of new knowledge, which often makes the content of a case base partially obsolete. This article proposes to make use of the concept of prototypical case to ensure that a CBR system would keep up-to-date with current research advances in the biomedical field. It proposes to illustrate and discuss the different roles that prototypical cases can serve in biomedical CBR systems, among which to organize and structure the memory, to guide the retrieval as well as the reuse of cases, and to serve as bootstrapping a CBR system memory when real cases are not available in sufficient quantity and/or quality. This paper presents knowledge maintenance as another role that these prototypical cases can play in biomedical CBR systems.

1 Introduction

Case-based reasoning is a valued knowledge management methodology in biomedical domains because it finds its recommendations on contextual knowledge. This type of knowledge is much more detailed and to the point for solving clinical problems, and allows to account for some of the complexity inherent to working in clinical domains. If the value of contextual, instance-based knowledge, is not in question, main issues for CBR methodology are how to keep up with the rate of generation of new biomedical knowledge, and how to maintain the recency of the knowledge represented as cases in a case base [21]. The system presented here proposes to automate the process of maintaining the recency of the knowledge represented in cases through maintenance prototypical cases, which can be mined from current biomedical literature. In the system presented in this article, prototypical cases serve as a structuring mechanism for the case-based reasoning, the case base being organized around them. They also guide the different steps of the reasoning process, for example the retrieval and the reuse. During reuse, current medical recommendations, represented in these prototypical cases mined from biomedical literature, guide the reuse of past cases and automatically revise obsolete recommendations from past cases.

Knowledge Base - Pathways

Print Search Help

Pathway name:

Snomed code: Category:

ADD MOD DEL FINDINGS

(JaundiceNOS No M (MediumImportance) AmMS ;
 OR Nausea No M (MediumImportance) AmMS ;
 OR Anorexia No M (MediumImportance) AmMS ;
 OR Malaise No M (MediumImportance) AmMS ;
 OR TemperatureIncreased No M (MediumImportance) AmMS ;
 OR PainNOS No M (MediumImportance) AmMS ; site = RightUpperQuadrantAbdomen
 OR StoolSymptom No M (MediumImportance) AmMS ; color = light
 OR UrinarySystemSignsAndSymptoms No M (MediumImportance) AmMS ; site = urine AND color = dark
 OR Hepatomegaly No M (MediumImportance) AmMS ;
 OR Ascites No M (MediumImportance) AmMS ;
 OR PeriothelialEdema No M (MediumImportance) AmMS ;

ADD MOD DEL DIAGNOSIS ASSESSMENT

HepaticFunctionPanel C (Compulsory) 1 ; AlkalinePhosphatase = Elevated OR AST = Elevated OR ALT = Elevated
 AND HepatitisPanelMeasurement H (High) 1 ; result = negative
 AND UltrasonographyAbdomenNOS(USNABD) H (High) 1 ; finding = Normal
 AND CBC, DIFFERENTIAL AND PLATELET COUNT H (High) 1 ; Eosinophils = Elevated
 IF HepatitisCAntigenMeasurement ; result = Positive ;
 AND HCVMeasurement H (High) 2 ; finding = Negative
 IF HepatitisBAntigenMeasurement ; result = Positive ;
 AND HBVDNAMeasurement H (High) 2 ; finding = Negative
 AND OralExamination M (MediumImportance) 1 ; finding = Abnormal

ADD MOD DEL TREATMENT / SOLUTION

IF ImmunosuppressantAgentNOS ; state = Absent ;
 StartPrednisoneAndCyclosporineTherapy H (High) 1 ;
 IF ImmunosuppressantAgentNOS ; state = Present ;
 StartAndFollowSalvageTreatmentProtocol H (High) 1 ;
 DiscontinueHepatotoxicDrugs M (MediumImportance) 1 ;
 IF PDN ; state = Present ;AND Patient ; condition = Stable ;
 StartAndFollowUDCATreatmentProtocol M (MediumImportance) 1 ;

Retrieving pathways for name = LiverChronicGVHD...

Fig. 1. A prototypical case, called here a clinical pathway, for liver chronic graft versus host disease (CGVHD)

The prototypical case structure adopted here is the one chosen for the Mémoire project, which is presented in the next section. The third section explains the role of case-based knowledge to represent contextual knowledge in biomedicine. The fourth section summarizes how prototypical cases can capture latest advances in biomedical literature, through a text mining mechanism. The fifth section presents how prototypical cases can serve as preserving the currency of a case base. A detailed example is presented in the sixth section. It is followed by an evaluation, a discussion, and a conclusion.

2 Mémoire Project

The goal of the Mémoire project [7] at the University of Washington is to provide a framework for the creation and interchange of cases, concepts, and CBR systems in biology and medicine.

The cornerstone of the knowledge acquisition process has been the conception of prototypical cases, called clinical pathways in this system. This prototypical case structure has been proposed in *Mémoire* as a generic prototypical case representation structure [7]. The clinical pathways, 91 of them having been implemented in a previous test version of the system, correspond to clinical diagnostic categories for the most part, some of them corresponding also to essential signs and symptoms requiring specific assessment or treatment actions. The clinical pathways are knowledge structures represented from a domain ontology, namely: all diseases, functions (also known as signs and symptoms), labs, procedures, medications, sites, and planning actions. Most of the terms naming these objects are standardized using the Unified Medical Language System (UMLS) terminology [15]. Only the terms not corresponding to objects in the UMLS have been added to the domain specific ontology. In particular, the planning actions used in the Treatment part of a prototypical case did not exist in the UMLS and were all created for the system.

An example of a prototypical case is provided in Fig. 1. It shows that a prototypical case, mostly a diagnostic category or disease, such as here chronic graft versus host disease affecting the liver, which is a complication of stem-cell transplantation, comprises three parts:

1. A list of *findings*, corresponding to signs and symptoms.
2. A *diagnosis assessment plan*, which is a plan to follow for confirming (or informing) the suspected diagnosis.
3. A *treatment plan*, which is a plan to follow for treating this disease when confirmed, or a solution when the pathway does not correspond to a disease.

The diagnosis assessment part and the treatment part can also be seen as simplified algorithms, since they use IF-THEN-ELSE structures, and LOOP structures, as well as SEQUENCE structures of actions in time. When instantiated with actual patients' data, this knowledge structure allows for sophisticated adaptation.

3 Cases as Contextual Knowledge

The gold standard for evaluating the quality of biomedical knowledge relies on the concept of evidence. Pantazi et al. propose an extension of the definition of *biomedical evidence* to include knowledge in individual cases, suggesting that the mere collection of individual case facts should be regarded as evidence gathering [16] (see Fig. 2). To support their proposal, they argue that the traditional, highly abstracted, hypothesis centric type of evidence that removes factual evidence present in individual cases, implies a strong *ontological commitment* to methodological and theoretical approaches, which is the source of the never-ending need for *current* and *best* evidence, while, at the same time, offering little provisions for the reuse of knowledge disposed of as obsolete. By contrast, the incremental factual evidence about individuals creates, once appropriately collected, a growing body of context-dependent evidence that can be reinterpreted and reused as many times as possible.

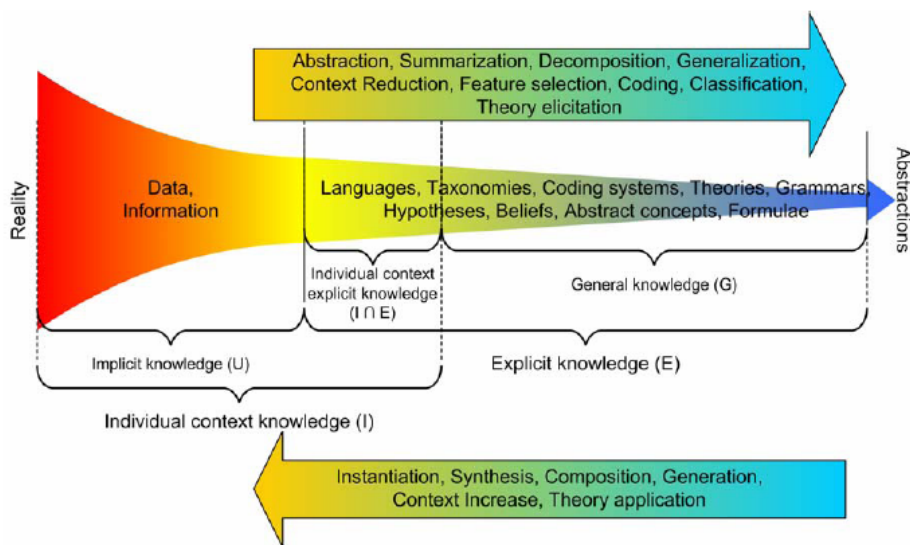


Fig. 2. The knowledge spectrum in biomedical informatics [16]

Currently, the concept of evidence most often refers to an abstract proposition derived from multiple, typically thousands of cases, in the context of what is known as a *randomized control trial*. Hypothesis forming is the cornerstone of this kind of biomedical research. Hypotheses that pass an appropriately selected statistical test become evidence. However, the process of hypothesis forming also implies a commitment to certain purposes (e.g., research, teaching, etc.), and inherently postulates ontological and conceptual reductions, orderings and relationships. All these are direct results of the particular conceptualizations of a researcher that is influenced by experience, native language, background, etc. This reduction process will always be prone to errors as long as uncertainties are present in our reality. In addition, even though a hypothesis may be successfully verified statistically and may become evidence subsequently, its applicability will always be hindered by our inability to fully construe its complete meaning. This meaning is fully defined by the complete context where the hypothesis was formed and which include the data sources as well as the context of the researcher that formed the hypothesis.

The discussion about commitment to research designs, methodological choices, and research hypotheses led Pantazi et al. to the proposal to extend the definition and the understanding of the concept of evidence in biomedicine and align it with an intuitively appealing and an important direction of research: *Case-Based Reasoning* (CBR) [17]. From this perspective, the concept of evidence, traditionally construed on the basis of knowledge applicable to populations, is evolved to a more complete, albeit more complex construct which emerges naturally from the attempt to understand, explain and manage unique, individual cases. This new perspective of the concept of evidence is surprisingly congruent with the current acceptance of the notion of evidence in forensic science for instance. Here, by evidence, one also means, besides general patterns and trends that apply generally to populations, the recognition of any

spatio-temporal form (i.e., pattern, regularity) in the spatio-temporal context of a case (e.g., a hair, a fibre, a piece of clothing, a smell, a fluid spot, a sign of struggle, a finger print on a certain object, the reoccurrence of a certain event, etc.) and which may be relevant to the solution to that case. This new view where a body of evidence is incremental in nature and accumulates dynamically in form of facts about individual cases is a striking contrast with traditional definitions of biomedical evidence. In addition, case evidence, once appropriately collected, represents a history that can be reinterpreted and reused as many times as necessary. But most importantly, the kind of knowledge where the “what is”, i.e., case data, is regarded as evidence can be easily proven to be less sensitive to the issues of *recency* (i.e., current evidence) and *validity* (i.e., best evidence).

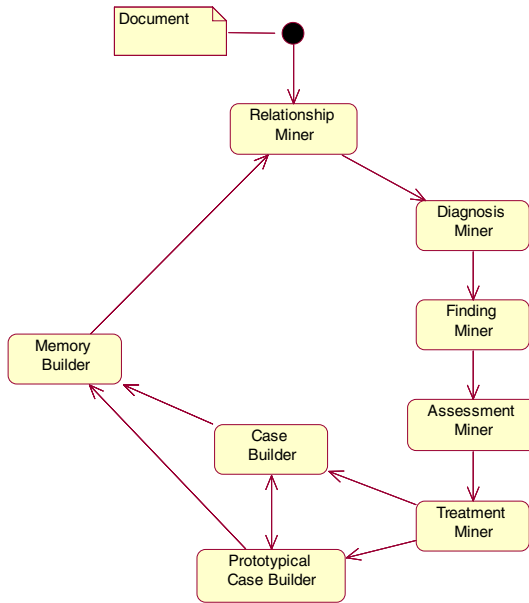


Fig. 3. ProConceptMiner architecture

If the question of up-to-date or current knowledge is not as critical for cases as for general knowledge, it is nevertheless an interesting research question to study how to keep the content of a case base current. Biomedical procedures, tests, and practices change, while recorded cases do not. This article proposes prototypical cases as a media for merging current and alternate medical practice with the highly context specific content of a case base.

4 Prototypical Case Mining

ProCaseMiner system (see Fig. 3) mines for cases and prototypical cases from biomedical literature [8]. A selection of documents for a given medical domain is the

input to this system. Pertinent documents may be literature articles, but also textual clinical practice guidelines, and medical case studies. It is important that such documents should all be related to a given domain, such as in our example stem-cell transplantation.

ProConceptMiner core component is the RelationshipMiner, which mines for triples $\langle \text{concept-1}, \text{relationship-1,2}, \text{concept-2} \rangle$ from a document. It also attaches a condition to a triple when it finds it to represent the information that IF a condition occurs, then an action or test is undertaken. This can be represented as $\langle \text{concept-1}, \text{relationship-1,2}, \text{concept-2} \rangle$ IF $\langle \text{concept-3}, \text{relationship-3,4}, \text{concept-4} \rangle$. An example can be $\langle \text{Patient}, \text{startTreatment}, \text{PrednisoneAndCyclosporineTherapy} \rangle$ IF $\langle \text{absent}, \text{property_of}, \text{ImmunosuppressantAgentNOS} \rangle$. This structure is called a triple pair.

ProConceptMiner interprets the results from RelationshipMiner by successively mining for diagnoses in DiagnosisMiner, findings in FindingMiner, assessments in AssessmentMiner, and treatments in TreatmentMiner. Following, it builds cases from these results in CaseBuilder or PrototypicalCaseBuilder. The order between these two components can be altered since in some cases, learnt relationships will be associated with conditions, which signals a prototypical case, and in others there will not be any of these conditions, which signals a practice case. Generally, from medical articles and clinical practice guidelines, the learnt artifact will be a prototypical case. From clinical case studies, the learnt artifact will be a practice case. The previous steps deal with prototypical cases and practice cases built from scratch from a single document. A next step is to consolidate learning results across documents. This step is called MemoryBuilder [8].

5 Prototypical Cases for Knowledge Maintenance

Mémoire system relies on a generic prototypical case representation to perform its case-based reasoning and to maintain the recency of its knowledge.

5.1 Case Representation

The elements of the representation language are those of semantic networks:

- A *domain ontology*, which is the set of *class symbols* (also called concepts in the UMLS [15]) C , where C_i and C_j denote elements of C . Specific subdomains are for example findings (signs and symptoms, noted F_i), tests and procedures (A_i), and planning actions (P_i).
- A *set of individual symbols* (also called instances) I , where i and j denote elements of I . Among these, some refer to instances of classes, others to numbers, dates, and other values. Instances of a class C_i are noted aC_i .
- A *set of operator symbols* O , permits to form logical expressions composed of classes, instances and other values, and relationships. Prototypical cases and clinical cases are expressed this way, and such a composition permits to represent complex entities in a structured format. The set of operators comprises the following:

\wedge (AND)
 \vee (OR)
 ATLEAST n
 ATMOST n
 EXACTLY n
 IF

Prototypical cases are expressed as $\langle \text{problem situation}, \text{solution} \rangle$, where *problem situation* is expressed in the object-oriented knowledge representation language above as a composition of instances with operators and where *solution* also has the same representation, but adds other operators to express conditional expressions (*IF*):

$$\begin{aligned}
 \text{problem situation} &= \Theta aF_i \{ \langle \text{att}_i, \text{val}_i \rangle \} \\
 \text{solution} &= \Theta aA_i \{ \langle \text{att}_i, \text{val}_i \rangle \} \\
 &\quad \Theta aP_i \{ \langle \text{att}_i, \text{val}_i \rangle \}
 \end{aligned}$$

with for prototypical cases: $\Theta \in O$, the default value being \vee for prototypical cases, and for clinical cases: $\Theta \in \{ \wedge \}$, the default value being \wedge for clinical cases.

The default representation for clinical cases is the same as for prototypical cases, except that the only connector available here is the connector \wedge both for problem situation and for solution. Since a case is not abstracted, cases are expressed using only \wedge .

5.2 Memory Organization

The memory of the system is organized in several layers, where the prototypical cases index the clinical cases (see Fig. 4). Several kinds of prototypical cases may be available:

- The *expert prototypical cases*, which were provided by the experts when the system was built. The roles of these cases are to provide a structure to the memory, and to organize the clinical/experiential cases so as to facilitate the search through the memory.
- The *maintenance prototypical cases*, which provide the updates coming from the literature. These may be reviewed by humans as well – regular staff or experts. The role of these cases is to maintain the knowledge represented in the case base.
- The *learnt prototypical cases*, which are learnt through conceptual clustering from the cases that enrich the memory over time [6]. These prototypical cases have for main role to facilitate the search through the memory, as well as a role of suggesting research questions [5].

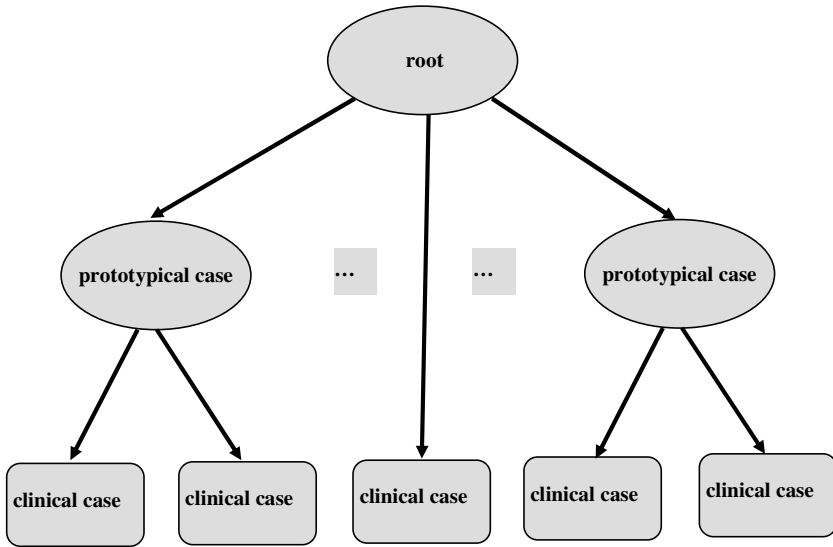


Fig. 4. Memory organization

5.3 Reasoning Process

The reasoning process starts with the presentation to the system of a new problem to solve. This system is capable of handling the wide variety of problems that physicians can face when they take care of patients, and the first task of the system is to determine the nature of the problem to solve. Classically, the reasoning of the system proceeds through the following steps [1]:

- [1] **Interpretation:** given the description of a patient problem, the system constructs, by interpretation, the initial situation expressed in the knowledge representation language of the system. Abstraction is the main reasoning type used here, and in particular temporal abstraction to create trends from time-stamped data. Numerical values are abstracted into qualitative values. Let c_c be the target patient case to solve, represented as a conjunction of findings:

$$c_c = \Theta aF_i \{ \langle att_i, val_i \rangle \}$$

- [2] **Prototype-guided retrieval:** the case-base is searched for prototypical cases and cases matching this new problem to solve through case-based retrieval. The result is a set containing both cases and prototypical cases. Let CS be this conflict set: $CS = \{ c_i, p_j \}$ where the c_i are cases and the p_j are prototypical cases. Cases are only retrieved directly if they are not indexed under a prototypical case – which is rare.
- [3] **Conflict resolution (R_r):** the following hierarchy of reuse is followed:
- I. reuse expert prototypical cases
 - II. reuse maintenance prototypical cases
 - III. reuse learnt prototypical cases
 - IV. reuse cases

Nevertheless, the first criterion to choose the entity to reuse is the number of problem description elements matched. The entities are ranked by decreasing number of matched problem description elements with the target case to solve.

Most of the time, the most similar entity is a prototypical case. The retrieval is then guided by the prototypical case(s) ranked higher, and the clinical cases indexed under this prototypical case are retrieved and ranked by decreasing similarity with the target case to solve. If no clinical case is available, the prototypical case will be reused.

- [4] **Prototype-guided reuse:** the reuse of a prototypical case entails evaluating the preconditions of any IF-THEN statement and keeping only those that are satisfied and selecting them in an ordered manner directed by the *order* attribute attached to each assessment or treatment class. The reuse of a clinical case is guided by the expert or maintenance prototypical case in such a manner that the prototypical case can substitute, add, or delete recommendations from the case.
- [5] **Retain:** when the solution is complete, and after feedback from the application, it is memorized with the target case solved.

The system provides a list of recommendations represented as instances of assessment and/or planning actions.

6 Example

This section presents an example of prototypical case guided retrieval and reuse. A patient consults his doctor about new symptoms occurring after his transplant. The patient's symptoms are: *Nausea*, *Malaise*, and *PainNOS* localized in the upper right portion of the abdomen. The physician records the main complaint of the patient, which is the unusual abdominal pain.

The physician reviews the drugs the patient is taking, as well his chart with the latest labs and physical exams. The patient is not taking any immunosuppressant drug, nor any hepato-toxic drug.

6.1 Prototype Guided Retrieval

The three symptoms of the patient each trigger several prototypical cases:

- *Nausea* triggers 18 prototypical cases: *LiverChronicGVHD*, *HepatitisAcuteNOS*, *LiverDrugToxicity*, *GastricChronicGVHD*, *GastricHemorrhage*, *ColonChronicGVHD*, *DrugInducedNauseaAndVomiting*, *DuodenalChronicGVHD*, *EsophagealChronicGVHD*, *EsophagealInfection*, *IntestinalDrugToxicity*, *AdrenalInsufficiency*, *UrethralInfection*, *BladderInfection*, *RecurrentNonHodgkin'sLymphoma*, *NonInfectiousPericarditisNOS*, *AcuteCholecystitis*, and *Hypomagnesemia*.
- *Malaise* triggers 4 prototypical cases: *LiverChronicGVHD*, *HepatitisAcuteNOS*, *LiverDrugToxicity*, and *InfectiousMononucleosis*.

- *PainNOS* in *RightUpperQuadrant* triggers 4 prototypical cases: *LiverChronicGVHD*, *LiverDrugToxicity*, *HepatitisAcuteNOS*, and *AcuteCholecystitis*.

The similarity measure ranks highest *LiverChronicGVHD* and *HepatitisAcuteNOS*, because *LiverDrugToxicity* is ruled out by the fact that the patient is not taking any hepato-toxic drug. Therefore the cases chosen to base the reuse are: *LiverChronicGVHD* (see Fig. 5 and Fig. 6 for this prototypical case) and *HepatitisAcuteNOS*.

In this particular example, the system does not retrieve the cases indexed under these prototypical cases because all the features describing the case to solve are accounted for in the prototypical cases. Most of the time though the actual clinical cases would be retrieved, since they would often match some of the features not present in a prototypical case.

6.2 Prototype Guided Reuse

The reuse in this case combines the diagnosis assessment and eventually the treatment plan of two prototypical cases: *LiverChronicGVHD* and *HepatitisAcuteNOS*.

The diagnosis assessment proceeds in four stages, as indicated by the range of *order* in the *LiverChronicGVHD* prototypical case (from 1 to 4, rightmost column in Fig. 5 and Fig. 6).

- First, request a *HepaticFunctionPanel*, a *HepatitisPanel*, an *UltrasonographyAbdomenNOS*, and a *CBC*. The first steps of diagnosis assessment for both *LiverChronicGVHD* and *HepatitisAcuteNOS* being the same, the system does not propose any additional procedures to be performed at first.
- Second, after the results have come in, and if they have the values indicated in the case, proceed with *HCVRNAMEasurement* if *HepatitisCAntigenMeasurement* was positive, and with *HBVDNAMEasurement* if *HepatitisBAntigenMeasurement* was positive. The patient tested negative to hepatitis, therefore these procedures are not requested.
- Third, request an *OralExamination*, a *BiopsyOfLipNOS*, a *BiopsyOfSkinNOS*, and a *SchirmerTearTest*. The patient undertook all of these, and tested positive for *SkinChronicGVHD* in his lip biopsy.
- Fourth, because the patient tested positive for *SkinChronicGVHD*, he will not have to undergo *BiopsyOfLiver*, and his diagnosis of *LiverChronicGVHD* is established.

The treatment plan starts in this prototypical case only after the diagnosis is established because of the order of 1 indicated in the rightmost column of the treatment plan (see Fig. 6). If the order had been 0, some treatment would have started just by triggering this case. Since the patient is not taking any immunosuppressant drug, he will be placed on prednisone and cyclosporine therapy (*StartPDNCSPTtherapy*). The other actions are eliminated because their preconditions are not met (for the second one), and not yet met (for the third one). After some time, if the patient is considered as stable, the third statement will be considered: *ConsiderUDCARxProtocol*.

GastrointestinalDiagnoses : LiverChronic GVHD (-----)

Findings

Importance: N (NecessaryAndSufficient) C (Compulsory), H (High Importance), M (MediumImportance), L (LowImportance), S (SecondaryImportance) [default = 'M'].
Level: A (Absent), m (Mild), M (Moderate), S (Severe) [default = 'AmMS'].

Connector	Finding Name	Snomed code	(Properties, Values)	Importance	Level
	(JaundiceNOS	M-57610		H	
OR	Nausea	F-52760		M	
OR	Anorexia	F-50015		M	
OR	Malaise	F-01220		M	
OR	Fever	F-03003		M	
OR	PainNOS	F-A2600	site=RightUpperQuadrantAbdomen	M	
OR	Stool	T-59666	color=light	M	
OR	Urine	T-70060	color=dark	M	
OR	Hepatomegaly	D5-81220		M	
OR	Ascites	D5-70400		M	
OR	PeripheralEdema)	M-36330		M	
AND	HepatoToxicDrug			H	A

Diagnosis Assessment

Connector	Procedure Name	Snomed code	(Properties, Values)	Importance	Order
	HepaticFunctionPanel	P3-09100	finding=AlkalinePhosphataseMeasurement(ALKP)(P3-71350) result=elevated OR finding=ASTMeasurement(AST)(P3-72000) result=elevated OR finding=ALTMeasurement(ALT)(P3-71220) result=elevated OR finding=LDHMeasurement(LDH)(P3-73380)result=elevated	C	1
AND	HepatitisPanel	P3-09110	finding=HepatitisPanelMeasurement(P3-64000) result=negative	H	1
AND	UltrasonographyAbdomenNO S(USNABD)	P5-BB200	finding=Normal	H	1
AND	CBC	P3-30100	Finding = Eosinophils result = elevated	H	1
IF HepatitisC AntigenMeasurement(P3-64054).result = Positive	HCVRNAmMeasurement	P3-64050	finding=negative AND synonym=HCVMeasurement	H	2
IF HepatitisB AntigenMeasurement(P3-64021) result=Positive	HBVDNAmMeasurement		finding=negative	H	2

Fig. 5. Liver ChronicGVHD (part I) prototypical case representation with its list of findings (corresponding to diagnoses), and its list of diagnosis assessment steps

7 Evaluation

A formal evaluation of the approach followed by Memoire can be found in CARE-PARTNER decision-support performance [7]. On 163 different clinical situations or cases, corresponding to contacts between the system and a clinician about three patients, the system was rated 82.2% as *Meets all standards*, and 12.3% as *Adequate*, for a total of 94.5% of results judged clinically acceptable by the medical experts. The advice provided by the system covers most of the clinicians’ tasks: labs and procedure results interpretation, diagnosis assessment plan, treatment plan, and pathways information retrieval. Pathways represent prototypical cases retrieved by the system, and

correspond to diagnostic categories (see Fig. 1 for an example). Important in this system is the evolution of the competency of the system over time, reaching 98.6% *Meets all standards/Adequate* for patient 3 for all his 54 contacts.

8 Discussion

Some may object to the need of maintenance prototypical cases from the literature, stating that a case base will naturally evolve into a more current one by adding newly solved cases over time. Actually this is an important advantage of case-based reasoning to constantly learn and improve its case-based knowledge over time in an incremental manner. Nevertheless, from the experience of Carepartner system [7], the availability of clinical guidelines is considered as a required standard of care in a medical domain. They represent the level of care to which clinicians are legally required to abide. It is therefore essential for CBR in biomedicine to provide a mechanism to infuse clinical guideline based knowledge within case-based recommendations.

In CBR research, generalized cases are named in varied ways, such as prototypical cases, abstract cases, prototypes, stereotypes, templates, classes, categories, concepts, and scripts – to name the main ones [13]. Although all these terms refer to slightly different concepts, they represent structures that have been abstracted or generalized from real cases either by the CBR system, or by an expert. When these prototypical cases are provided by a domain expert, this is a knowledge acquisition task [3]. More frequently, they are learnt from actual cases. In CBR, prototypical cases are often learnt to structure the memory.

Many authors mine for *prototypes*, and simply refer to *induction* for learning these. CHROMA [2] uses induction to learn prototypes corresponding to general cases, which each contain a pair $\langle \textit{situation}, \textit{plan} \rangle$, where the situation is an object whose slots have several values possible – values are elements of a set. Bellazzi et al. organize their memory around prototypes [4]. The prototypes can either have been acquired from an expert, or induced from a large case base. Schmidt and Gierl point that prototypes are an essential knowledge structure to fill the gap between general knowledge and cases in medical domains [14]. The main purpose of this prototype learning step is to guide the retrieval process and to decrease the amount of storage by erasing redundant cases. A generalization step becomes necessary to learn the knowledge contained in stored cases. They use several threshold parameters to adjust their prototypes, such as the number of cases the prototype is filled with, and the minimum frequency of each contraindication for the antibiotic therapy domain [20].

Others specifically refer to *generalization*, so that their prototypes correspond to generalized cases. An example of system inducing prototypes by generalization is a computer aided medical diagnosis system interpreting electromyography for neuropathy diagnosis [12]. The first prototypes are learnt from the expert by supervised learning, then the prototypes are automatically updated by the system by generalizing from cases. Prototypes can fusion if one is more general than the other ones, or new prototypes can be added to the memory. Portinale and Torasso in ADAPTER organize their memory through E-MOPs learnt by generalization from cases for diagnostic problem-solving [19]. E-MOPs carry the common characteristics of the cases they

index, in a discrimination network of features used as indices to retrieve cases. Mougouie and Bergmann present a method for learning generalized cases [14]. This method, called the Topkis-Veinott method, provides a solution to the computation of similarity for generalized cases over an n -dimensional Real values vector. Maximini et al. have studied the different structures induced from cases in CBR systems [13]. They point out that several different terms exist, such as generalized case, prototype, schema, script, and abstract case. The same terms do not always correspond to the same type of entity. They define three types of cases. A point case is what we refer to as a real case. The values of all its attributes are known. A generalized case is an arbitrary subspace of the attribute space. There are two forms: the attribute independent generalized case, in which some attributes have been generalized (interval of values) or are unknown, and the attribute dependent generalized case, which cannot be defined from independent subsets of their attributes.

Yet other authors refer to *abstraction* for learning abstract cases. Branting proposes case abstractions for its memory of route maps [9]. The abstract cases, which also contain abstract solutions, provide an accurate index to less abstract cases and solutions. [18] learns prototypes by abstracting cases as well.

Finally, many authors learn *concepts* through *conceptual clustering*. MNAOMIA [5, 6] learns concepts and trends from cases through *conceptual clustering* similar to GBM [11]. Perner learns a hierarchy of classes by *hierarchical conceptual clustering*, where the concepts represent clusters of prototypes [18].

Díaz-Agudo and González-Calero use *formal concept analysis* (FCA) – a mathematical method from data analysis – as another induction method for extracting knowledge from case bases, in the form of *concepts* [10].

The system presented here also uses prototypical cases to organize its memory, direct its retrieval and its adaptation. Its originality lies in reusing both clinical cases and prototypical cases, judiciously combining their recommendations to build more up-to-date recommendations. The prospect of using prototypical cases for case base maintenance is also novel, even in comparison with Schmidt and Gierl [14] whose maintenance is directed toward summarizing several cases, and not toward providing more current knowledge. In addition, the mining process for mining prototypical cases from the literature is also novel in CBR, and is explained in [8].

9 Conclusion

This system proposes to keep a case base up-to-date by automatically learning prototypical cases from biomedical literature. These prototypical cases are an important memory structure which the systems relies upon for guiding its retrieval and reuse steps. These prototypical cases, called maintenance prototypical cases, provide a method for enabling a case base to naturally evolve and follow the otherwise overwhelming flow of biomedical advances. Coupled with the concept of mining prototypical cases from biomedical literature, this methodology moves a step forward in the direction of automatically building and maintaining case bases in biomedical domains. Future areas of research are to study how prototypical cases learnt from clinical cases, from the experts, and from the literature can complement one another, and

how the reasoner can take advantage of the knowledge provided by each in a harmonious and advantageous way.

References

- [1] Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodologies Variations, and Systems Approaches. *AI Communications*, IOS Press 7(1), 39–59 (1994)
- [2] Armengo, E., Plaza, E.: Integrating induction in a case-based reasoner. In: Haton, J.-P., Manago, M., Keane, M.A. (eds.) *Advances in Case-Based Reasoning*. LNCS, vol. 984, pp. 243–251. Springer, Heidelberg (1995)
- [3] Bareiss, R.: *Exemplar-Based Knowledge Acquisition*. Academic Press, San Diego (1989)
- [4] Bellazzi, R., Montani, S., Portinale, L.: Retrieval in a Prototype-Based Case Library: A Case Study in Diabetes Therapy Revision. In: Smyth, B., Cunningham, P. (eds.) *EWCBR 1998*. LNCS (LNAI), vol. 1488, pp. 64–75. Springer, Heidelberg (1998)
- [5] Bichindaritz, I.: A case based reasoner adaptive to several cognitive tasks. In: Aamodt, A., Veloso, M.M. (eds.) *Case-Based Reasoning Research and Development*. LNCS, vol. 1010, pp. 391–400. Springer, Heidelberg (1995)
- [6] Bichindaritz, I.: Case-Based Reasoning and Conceptual Clustering: For a Co-operative Approach. In: Watson, I.D. (ed.) *Progress in Case-Based Reasoning*. LNCS, vol. 1020, pp. 91–106. Springer, Heidelberg (1995)
- [7] Bichindaritz, I.: Mémoire: Case-based Reasoning Meets the Semantic Web in Biology and Medicine. In: Funk, P., González Calero, P.A. (eds.) *ECCBR 2004*. LNCS (LNAI), vol. 3155, pp. 47–61. Springer, Heidelberg (2004)
- [8] Bichindaritz, I.: Prototypical Case Mining from Biomedical Literature. *Applied Intelligence* (in press) (2007)
- [9] Branting, K.L.: Stratified Case-Based Reasoning in Non-Refinable Abstraction Hierarchies. In: Leake, D.B., Plaza, E. (eds.) *ICCBR 97*. LNCS, vol. 1266, pp. 519–530. Springer, Heidelberg (1997)
- [10] Diaz-Agudo, B., González-Calero, P.: Classification Based Retrieval Using Formal Concept Analysis. In: Aha, D.W., Watson, I. (eds.) *ICCBR 2001*. LNCS (LNAI), vol. 2080, pp. 173–188. Springer, Heidelberg (2001)
- [11] Lebowitz, M.: Concept Learning in a Rich Input Domain: Generalization-Based Memory. In: Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (eds.) *Machine Learning: An Artificial Intelligence Approach*, vol. 2, pp. 193–214. Morgan Kaufmann, San Francisco (1986)
- [12] Malek, M., Rialle, V.: A Case-Based Reasoning System Applied to Neuropathy Diagnosis. In: Haton, J.-P., Manago, M., Keane, M.A. (eds.) *Advances in Case-Based Reasoning*. LNCS, vol. 984, pp. 329–336. Springer, Heidelberg (1995)
- [13] Maximini, K., Maximini, R., Bergmann, R.: An Investigation of Generalized Cases. In: Ashley, K.D., Bridge, D.G. (eds.) *ICCBR 2003*. LNCS, vol. 2689, pp. 261–275. Springer, Heidelberg (2003)
- [14] Mougouie, B., Bergmann, R.: Similarity Assessment for Generalized Cases by Optimization Methods. In: Craw, S., Preece, A.D. (eds.) *ECCBR 2002*. LNCS (LNAI), vol. 2416, pp. 249–263. Springer, Heidelberg (2002)
- [15] National Library of Medicine: The Unified Medical Language System. [Last access: 2005-04-01] (1995) <http://umls.nlm.nih.gov>
- [16] Pantazi, S.V., Arocha, J.F.: Case-based Medical Informatics. *BMC Journal of Medical Informatics and Decision Making* 4(1), 19–39 (2004)

- [17] Pantazi, S.V., Bichindaritz, I., Moehr, J.R.: The Case for Context-Dependent Dynamic Hierarchical Representations of Knowledge in Medical Informatics. In: Proceedings of ITCH '07 (in press) (2007)
- [18] Perner, P.: Different Learning Strategies in a Case-Based Reasoning System for Image Interpretation. In: Smyth, B., Cunningham, P. (eds.) EWCBR 1998. LNCS (LNAI), vol. 1488, pp. 251–261. Springer, Heidelberg (1998)
- [19] Portinale, L., Torasso, P.: ADAPTER: An Integrated Diagnostic System Combining Case-Based and Abductive Reasoning. In: Aamodt, A., Veloso, M.M. (eds.) Case-Based Reasoning Research and Development. LNCS, vol. 1010, pp. 277–288. Springer, Heidelberg (1995)
- [20] Schmidt, R., Gierl, L.: Experiences with Prototype Designs and Retrieval Methods in Medical Case-Based Reasoning Systems. In: Smyth, B., Cunningham, P. (eds.) EWCBR 1998. LNCS (LNAI), vol. 1488, pp. 370–381. Springer, Heidelberg (1998)
- [21] Wilson, D., Leake, D.B.: Mainting Case Based Reasoners: Dimensions and Directions. *Computational Intelligence Journal* 17(2), 196–213 (2001)

A Search Space Reduction Methodology for Large Databases: A Case Study

Angel Kuri-Morales¹ and Fátima Rodríguez-Eraza²

¹ Departamento de Computación. Instituto Tecnológico Autónomo de México

² Posgrado en Ciencias e Ingeniería de la Computación, Universidad Nacional Autónoma de México, Mexico

akuri@itam.mx, frodrigueze@uxmcc2.iimas.unam.mx

Abstract. Given the present need for Customer Relationship and the increased growth of the size of databases, many new approaches to large database clustering and processing have been attempted. In this work we propose a methodology based on the idea that statistically proven search space reduction is possible in practice. Two clustering models are generated: one corresponding to the full data set and another pertaining to the sampled data set. The resulting empirical distributions were mathematically tested to verify a tight non-linear significant approximation.

Keywords: Large databases, Sampling, Space reduction, Preprocessing, Clustering.

1 Introduction

Nowadays, commercial enterprises are importantly oriented to continuously improving customer-business relationship. With the increasing influence of CRM¹ Systems, such companies dedicate more time and effort to maintain better customer-business relationships. The effort implied in getting to better know the customer involves the accumulation of enormous data bases where the largest possible quantity of data regarding the customer is stored.

Data warehouses offer a way to access detailed information about the customer's history, business facts and other aspects of the customer's behavior. The databases constitute the information backbone for any well established company. However, from each step and every new attempted link of the company to its customers the need to store increasing volumes of data arises. Hence databases and data warehouses are always growing up in terms of number of registers and tables which will allow the company to improve the general vision of the customer.

Data warehouses are difficult to characterize when trying to analyze the customers from company's standpoint. This problem is generally approached through the use of data mining techniques [1]. However, to attempt direct clustering over a data base of several terabytes with millions of registers results in a costly and not always fruitful effort. There have been many attempts to solve this problem. For instance, with the

¹ Customer Relationship Management.

use of parallel computation, the optimization of clustering algorithms, via alternative distributed and grid computing and so on. But still the more efficient methods are unwieldy when attacking the clustering problem for databases as considered above.

In this article we present a methodology derived from the practical solution of an automated clustering process over large database from a real large sized (over 20 million customers) company. We emphasize the way we used statistical methods to reduce the search space of the problem as well as the treatment given to the customer's information stored in multiple tables of multiple databases.

Because of confidentiality issues the name of the company and the actual final results of the customer characterization are withheld.

Paper Outline

The outline of the paper is as follows. First, we give an overview of the analysis of large databases in section 2; next we give a clustering, sampling, and feature selection overview. In section 3 we briefly discuss the case study treated with the proposed methodology. Explanation of the methodology follows in Section 4. Finally, we concluded In Section 5.

2 Analysis of Large Databases

To extract the best information of a database it is convenient to use a set of strategies or techniques which will allow us to analyze large volumes of data. These tools are generically known as data mining (DM) which targets on new, valuable, and nontrivial information in large volumes of data. It includes techniques such as clustering (which corresponds to non-supervised learning) and statistical analysis (which includes, for instance, sampling and multivariate analysis).

2.1 Clustering in Large Databases

Clustering is a popular data mining task which consist of processing a large volume of data to obtain groups where the elements of each group exhibit quantifiably (under some measure) small differences between them and, contrariwise, large dissimilarities between elements of different groups. Given its importance as a very important data mining task, clustering has been the subject of multiple research efforts and has proven to be useful for many purposes [2].

Many techniques and algorithms for clustering have been developed, improved and applied [3], [4]. Some of them try to ease the process on a large database as in [5], [6] and [7]. On the other hand, the so-called "Divide and Merge" [8] or "Snakes and Sandwiches" [9] methods refer to clustering attending to the physical storage of the records comprising data warehouses. Another strategy to work with a large database is based upon the idea of working with statistical sampling optimization [10].

2.2 Sampling and Feature Selection

Sampling is a statistical method to select a certain number of elements from a population to be included in a sample. There exist two sampling types: probabilistic and nonprobabilistic. For each of these categories there exists a variety of sub

methods. The probabilistic better known ones include: a) Random sampling, b) Systematic sampling, and c) Stratified sampling. On the other hand the nonprobabilistic ones include methods such as convenience sampling, judgment sampling, and quota sampling. There are many ways to select the elements from a data set and some of them are discussed in [11]. This field of research, however, continues to be an open one [12], [13].

The use of sampling for data mining has received some criticism since there is always a possibility that such sampling may hamper a clustering algorithm's capability to find small clusters appearing in the original data [10]. However, small clusters are not always significant; such is the case of customer clusters. Since the main objective of the company is to find significant and, therefore, large customer clusters, a small cluster that may not be included in a sample is not significant for CRM.

Apart from the sampling theory needed to properly reduce the search space, we need to perform feature selection to achieve desirable smaller dimensionality. In this regard we point out that feature selection has been the main object of many researches [14], [15], and these had resulted in a large number of methods and algorithms [16]. One such method is "multivariate analysis". This is a scheme (as treated here) which allows us to synthesize a functional relation between a dependent and two or more independent variables. There are many techniques to perform a multivariate analysis. For instance, multivariate regression analysis, principal component analysis, variance and covariance analysis, canonical correlation analysis, etc., [17]. Here we focus on the explicit determination of a functional which maximizes the resulting correlation coefficient while minimizing its standard error. Clearly, this approach requires a sufficiently large number of models to consider, as will be discussed in the sequel.

3 Case Study

A data mining project was conducted for a very large multi-national Latin American company (one of the largest in Latin America) hereinafter referred to as the "Company". The Company has several databases with information about its different customers, including data about services contracted, services' billing (registered over a period of several years) and other pertinent characterization data. The Company offers a large variety of services to millions of users in several countries. Its databases are stored on IBM Universal Database version 7.0. In our study we applied a specific data mining tool (which we will refer to as "the miner") which works directly on the database. We also developed a set of auxiliary programs intended to help in data pre-processing.

The actual customer information that was necessary for the clustering process was extracted from multiple databases in the Company. Prior to the data mining process, the Company's experts conducted an analysis of the different existent databases and selected the more important variables and associated data related to the project's purpose: to identify those customers amenable to become *ad hoc* clients for new products under development and others to be developed specifically from the results of the study. Due to the variety of platforms and databases, such process of selection

and collection of relevant information took several months and several hundred man-hours.

The resulting database displayed a table structure that contains information about the characteristics of the customers, products or services contracted for the customer and monthly billing data over a one year period.

To test the working methodology the project teamed worked with a subset of 400,000 customers registers, consisting of a total of 415 variables divided in 9 data tables. Table 1 displays the characteristics of the data sources treated in this study.

Table 1. Data sources

Table	Columns	Rows	Description
TFB	25	400,000	Customer billing
TINT	121	400,000	Internet services
TPK	49	400,000	Data package services
TGRL	11	400,000	Customer's general data
TAC	2	73	Supply areas
TCC	2	4	Customer's credit rank code
TPA	3	183	Customer's permanence
TLPC	121	400,000	Local services
TSD	85	400,000	Digital services

Main Objective

As stated above, the main objective of the data mining project was to characterize the customers of the Company allowing in the near future - in accordance to customer characteristics - to offer new services and/or increase sales to existent or new customers.

4 Methodology

In order to apply a methodology whereupon the search space is efficiently and effectively reduced it is necessary to comply with several steps leading to the adequate representation and/or behavior of the data regardless of its primary origin. These steps are discussed in what follows.

- Data preprocessing
- Search space reduction
- Clustering

4.1 Data Preprocessing

This step included data cleaning by exhaustively searching for incomplete, inconsistent or missing data [18]. Additionally, we also had to transform non-numeric to numeric data. Resulting from this process unrecoverable registers were deleted. The number of such deleted records, however, was not significant.

From the original multiple-tables structure we defined a single-table view structure for which a process of denormalization was performed. This followed

from an analysis of the key-structure. In this view tables with the same key were merged and tables with different keys were included in the referenced tables as additional columns. The transformation resulted in a view with a structure with 415 attributes.

4.2 Search Space Reduction

To reduce the search space we work with the original data to obtain a sample which is not only a subspace but, rather, one that properly represents the original (full) set of data. We reduce the set both horizontally (reducing the number of tuples) and vertically (reducing the number of attributes) to obtain the “minable view”. Simultaneous reduction - horizontal and vertical - yields the smallest representation of the original data set. Vertical reduction is possible from traditional statistical methods, while horizontal reduction, basically, consists of finding the best possible sample. The following subsections discuss how we performed both reductions.

Vertical Reduction

To perform vertical reduction, multivariate analysis is required. There exist many methods to reduce the original number of variables. Here we simply used Pearson’s correlation coefficients. An exploration for correlated variables was performed over the original data. We calculate a correlation matrix for the 415 variables. We considered (after consulting with the experts) that those variables exhibiting a correlation factor equal or larger than 0.75 were redundant. Hence, from the original 415 variables only 129 remained as informationally interesting. In principle, out of a set of correlated variables only one is needed for clustering purposes. Which of these is to be retained is irrelevant; in fact, we wrote a program which simply performed a sequential binary search to select the (uncorrelated) variables to be retained.

Horizontal Reduction

This step is based on the hypothesis that a sample will adequately represent the full set of data. The size of the sample was determined at the offset by the Company’s experts; hence, 20% of the original data (after vertical reduction) was sampled. The elements of such sample were randomly (uniformly) selected. From the sample we validated the representation adequacy of this subset. A central issue to our work was the way the sample is validated. The process consists of the following steps:

1. Select several n equally sized samples. In our case $n = 5$.
2. Select sets of m variables to perform a goodness-of-fit test. We selected couples ($m=2$) of variables to prove that, within each sample, the behavior of the selected variables is statistically equivalent.
3. Perform a search for the best regressive function. To this effect we programmatically analyzed, in every case, 34 models (listed in table 2). From these we selected the one which displayed the highest Pearson correlation factor.
4. Perform steps 2 and 3 as long as there are more variables to evaluate.

Table 2. Evaluated regressive models

Family	Model	Equation
	Linear	$y = a + bx$
	Quadratic	$y = a + bx + cx^2$
	nth Order Polynomial	$y = a + bx + cx^2 + dx^3 + \dots$
Exponential Family	Exponential	$y = ae^{bx}$
	Modified Exponential	$y = ae^{b/x}$
	Logarithm	$y = a + b \ln x$
	Reciprocal Log	$y = \frac{1}{a + b \ln x}$
	Vapor Pressure Model	$y = e^{a+b/x+c \ln x}$
Power Law Family	Power	$y = ax^b$
	Modified Power	$y = ab^x$
	Shifted Power	$y = a(x - b)^c$
	Geometric	$y = ax^{bx}$
	Modified Geometric	$y = ax^{b/x}$
	Root	$y = ab^{1/x}$
	Hoerl Model	$y = ab^x x^c$
	Modified Hoerl Model	$y = ab^{1/x} x^c$
Yield-Density Models	Reciprocal	$y = \frac{1}{ax + b}$
	Reciprocal Quadratic	$y = \frac{1}{a + bx + cx^2}$
	Bleasdale Model	$y = (a + bx)^{-1/c}$
	Harris Model	$y = \frac{1}{(a + bx^c)}$
Growth Models	Saturation-Growth Rate	$y = \frac{ax}{b + x}$
	Exponential Association 2	$y = a(1 - e^{-bx})$
	Exponential Association 3	$y = a(b - e^{-cx})$

Table 2. (continued)

Family	Model	Equation
Sigmoidal Models	Gompertz Relation	$y = ae^{-e^{-bx}}$
	Logistic Model	$y = \frac{a}{1 + be^{-cx}}$
	Richards Model	$y = \frac{a}{(1 + e^{b-cx})^{1/d}}$
	MMF Model	$y = \frac{ab + cx^d}{b + x^d}$
	Weibul Model	$y = a - be^{-cx^d}$
Miscellaneous	Hiperbolic	$y = a + \frac{b}{x}$
	Sinusoidal	$y = a + b \cos(cx + d)$
	Heat Capacity	$y = a + bx + \frac{c}{x^2}$
	Gaussian Model	$y = ae^{-\frac{(x-b)^2}{2c^2}}$
	Rational Function	$y = \frac{a + bx}{1 + cx + dx^2}$

The following graphs illustrate the fact that several functions resulting from paired variables yield similar regressive fits. The data displayed in graphs 1a and 1b are closely adjusted with an MMF model; those of graphs 2a and 2b are, analogously, adjusted by a 4th degree polynomial; finally, the data displayed in graphs 3a and 3b are tightly fit by a rational function. Interestingly, the correlation coefficient in all three couples is better than 0.93 indicating the very high quality of the fit. Hence, we rest assured that all samples display statistically significant equivalence. (We note that, because of space limitations, we are unable to show the entire set; however, very similar remarks do apply in all cases). On the other hand, for different couples we obtain best fit with *different* models: MMF [(ab+cx^d)/(b+x^d)] for couple 1; 4th degree polynomial (a+bx+cx²+dx³+ex⁴) for couple 2 and a rational function [(a+bx)/(1+cx+dx²)] for couple 3. This fact reinforces our expectation that different variables distribute differently even though the samples behave equivalently. A hypothetical possibility which is ruled out from this behavior is that all variables were similarly distributed. If this were the case, then ALL models would behave similarly and no significant conclusion could be derived from our observations.

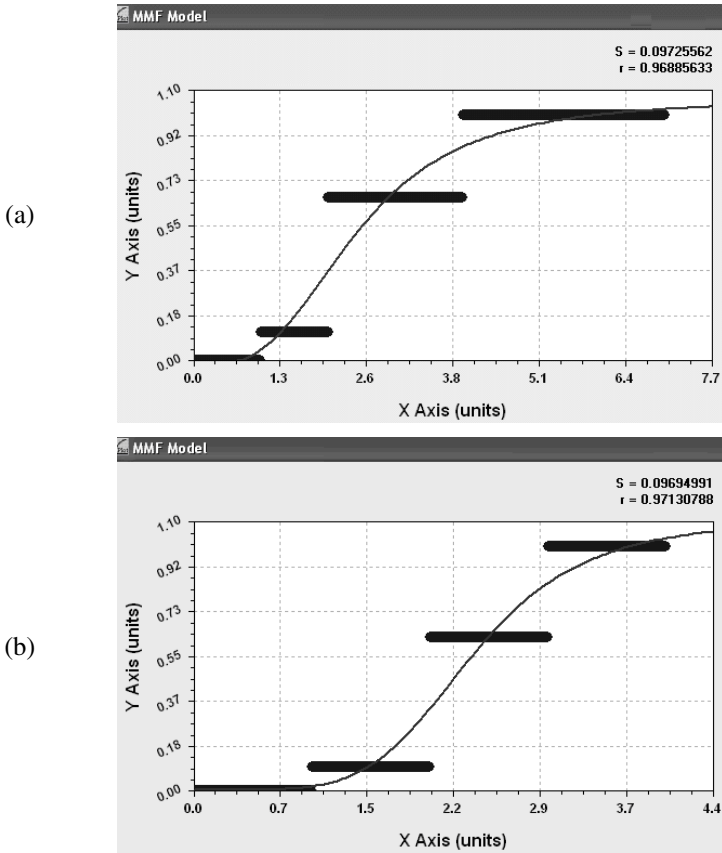


Fig. 1. Regressive fits. (a) MMF model for sample 1. (b) MMF model for sample 2.

It may be argued, upon first analysis, that the high correlation coefficients contradict the fact that our variables derive from the elimination of such correlation. Notice, however, that even if the variables with which we worked are not correlated (as discussed above) this non-correlation is *linear* (as pertaining to a Pearson coefficient) whereas the models considered here are basically highly non-linear, which resolves the apparent contradiction.

The probability of displaying results as shown by chance alone is less than 10^{-12} . We must stress the fact that this analysis is only possible because we were able to numerically characterize each of the subsets in 34 different forms and, thus, to select the most appropriate ones. Furthermore, not only characterization was proven; we also showed that, in every case, the said characterization was similar when required and dissimilar in other cases.

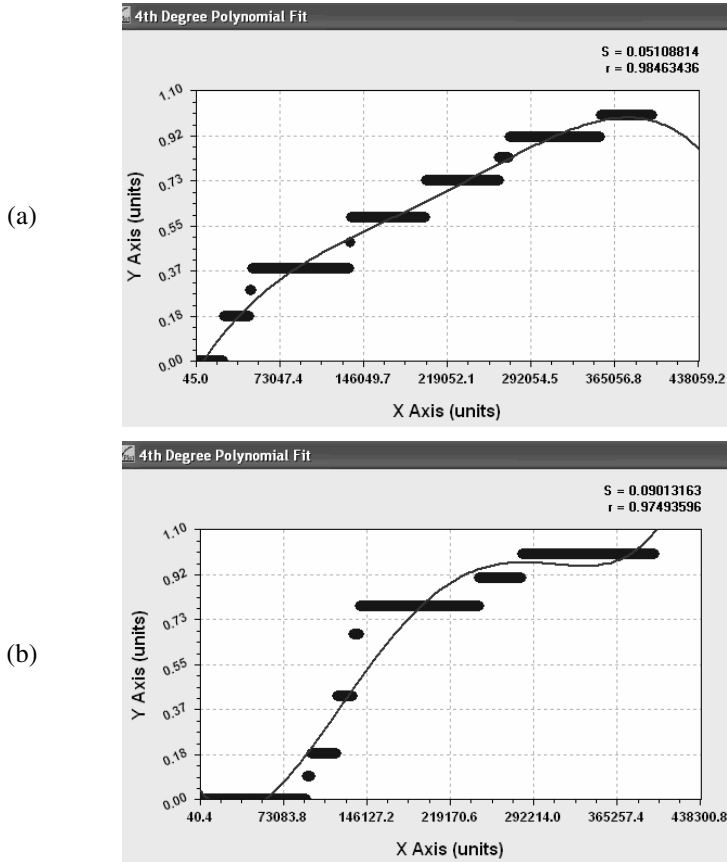


Fig. 2. Regressive fits. (a) 4th degree polynomial model for sample 1. (b) 4th degree polynomial model for sample 2.

4.3 Clustering Phase

Once the search space is reduced the clustering phase is reached. Before attempting the clustering proper, we impose certain a priori assumptions, as follows.

- The number of clusters is to be determined automatically (without applying any aprioristic rules).
- The “best” number (N) of clusters is derived from information theoretical arguments.
- The theoretical N is to be validated empirically from the expert analysis of the characteristics of such clusters.

In order to comply with our assumptions we follow the next steps:

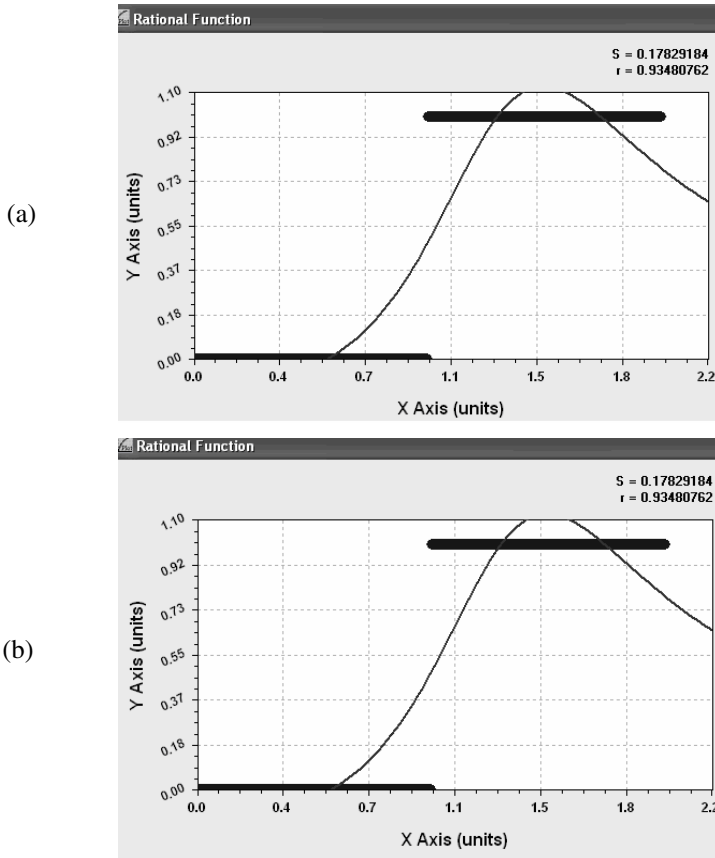


Fig. 3. Regressive fits. (a) Rational function model for sample 1. (b) Rational function model for sample 2.

1. Consecutively obtaining the clusters (via a Fuzzy C Means algorithm) assuming n clusters for $n=2, 3, \dots, k$; where “ k ” represents the largest acceptable number of clusters.
2. Determine the “optimal” number of clusters according to “elbow” criterion [10].
3. Clustering with a self organizing map algorithm to find the optimal segmentation.

The minable view with the 129 variables was processed. The Fuzzy C Means (FCM) algorithm was used on the uncorrelated data and the elbow criterion was applied [19]. It is important to stress the fact that the use of fuzzy logic allows us to determine the content of information (the entropy) in every one of the N clusters into which the data set is divided. Other clustering algorithms based on crisp logic do not provide such alternative. Since the elements of a fuzzy cluster belong to all clusters it is possible to establish an analogy between the membership degree of an element in the set and the probability of its appearance. In this sense, the “entropy” is calculated as the expected value of the membership for a given cluster. Therefore we are able to

calculate the partition’s entropy PE (see below). Intuitively, as the number of clusters is increased the value of PE increases since the structure within a cluster is disrupted. In the limit, where there is a cluster for every member in the set, PE is maximal. On the other hand, we are always able to calculate the partition coefficient: a measure of how compact a set is. In this case, such measure of compactness decreases with N. The elbow criterion stipulates that the “best” N corresponds to the point where the corresponding *tendencies* of PE to increase and PC to decrease *simultaneously* change. That is, when the curvature of the graph of tendencies changes we are faced with an optimal number of clusters. Table 3 displays part of the numeric data values of PC and PE. These coefficients were calculated with formulas 1 and 2.

$$PC = \sum_{k=1}^K \sum_{i=1}^c \frac{(\mu_{ik})^2}{K} \tag{1}$$

$$PE = -\frac{1}{K} \sum_{k=1}^K \sum_{i=1}^c \mu_{ik} \ln(\mu_{ik}) \tag{2}$$

Table 3. Numeric data for the elbow criterion

Clusters	2	3	4	5	6	7	8	9	10	11	12
PC	0.879	0.770	0.642	0.560	0.498	0.489	0.413	0.414	0.400	0.359	0.349
PE	0.204	0.436	0.639	0.812	0.982	1.036	1.220	1.224	1.272	1.403	1.433

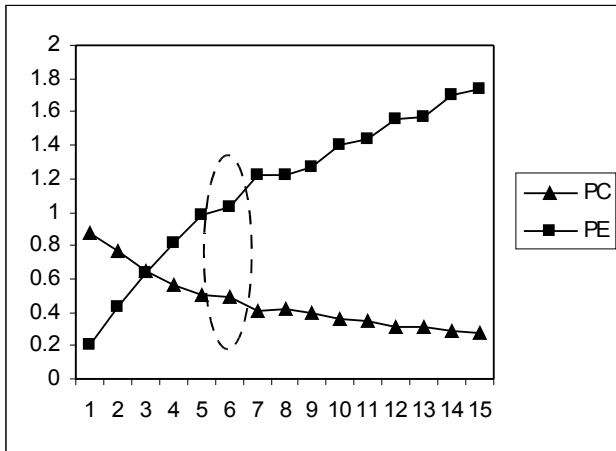


Fig. 4. Graph for the elbow criterion

Figure 4 shows the graph for the numeric data of table 3. In the graph the “elbow” point is located between the cluster 6 and 7, indicating that there is a high probability that the optimal number of clusters is in that point, i.e. N=6.

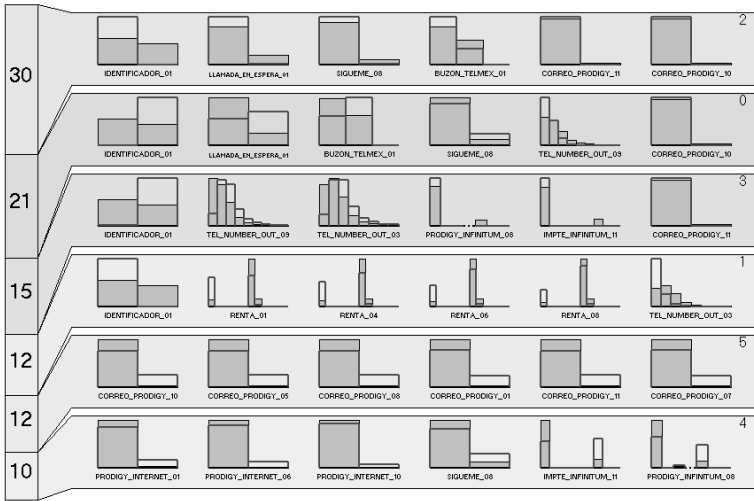


Fig. 5. Graph view of the clustering result supplied by the Miner

The last phase of our analysis implied the use of the miner and the theoretically determined best number of clusters, as shown in figure 5.

The graph shows at the left side the percentage of elements grouped in each cluster. On the right the neuron number which represents the cluster. Each cluster shows the more important variables for the results, ordered by Chi-squared characterization of the variable’s behavior in the cluster and in the whole sample. The cluster information for the Company can be extracted from the graph and reports supplied by the tool. We should now prove that clustering resulting from the reduced search space reflects a correct clustering view of the population.

4.4 Validation of the Reduced Search Space

To ease the understanding of the process in what follows we will call the clustering model from the sample “Model1”; likewise, we will call the clustering model derived from the complete data “Model2”.

We followed the next steps:

1. Reduce the original data set only vertically.
2. Execute a clustering process over the full set of data to obtain Model2.
3. Label all the original data set and the sample data set with Model1 and Model2.
4. Compare the resulting distribution of elements labeled with both models.

The results are discussed in what as follows.

Comparison of Model1 and Model2

Table 4 shows the percentages for the two models. The names of the clusters were replaced by letters to avoid possible confusions with the neuron numbers shown in the Miner’s results. As table 4 shows, the result clusters are very similar.

Table 4. Clusters' comparison for Model1 and Model2

Clusters	Model1 (%)	Model2 (%)	Difference (%)
A	30	27	3
B	21	20	1
C	15	18	3
D	12	15	3
E	12	12	0
F	10	8	2

Clustering from Sampling (Model1)

Having the clustering Model1, we labeled the sample data and the full data sets. The resulting distribution of elements into the different 6 clusters was expressed in percentages for comparison effects. As the table 5 shows the resulting distribution for the sample and for the full data are almost equal. This proves that the sample represents the full data set adequately.

Table 5. Labeling from the Model1 applied to the sampled and full data sets

Cluster	Sample (%)	Full Data (%)	Difference (%)
A	30.06	30.24	0.18
B	21.01	20.91	0.10
C	15.45	15.37	0.08
D	12.27	12.25	0.02
E	11.54	11.55	0.01
F	9.67	9.68	0.01

Cross Validation

Finally, we labeled the sample and the entire data with both algorithms. The results are shown in table 6.

Table 6. Comparison of Full and Sampled Data Clusters

Cl	Labeling derived from sampled Data				Labeling derived from Complete Data			
	Model1	Model2	Differ.	%Popu.	Model1	Model2	Differ.	%Popu.
A	23906	21503	2403	3%	120961	108084	12877	3%
B	16712	16155	557	1%	83655	81420	2235	1%
C	12285	14327	2042	3%	61471	72367	10896	3%
D	9760	11828	2068	3%	49013	59313	10300	3%
E	9179	9580	401	1%	46195	48356	2161	1%
F	7687	6136	1551	2%	38705	30460	8245	2%

As table 6 shows the differences between the distributions of elements into the clusters are similar between the two clustering models. Analog clusters share the same cardinality with a difference of less than 3%.

5 Conclusions

As we pointed out in the introduction, data mining may be an important strategic tool for commercial enterprises. But the management of large volumes of data (both physically and logically) may become a practical problem of large proportions and difficult to solve. Applying the methodology advanced herein it is possible to drastically reduce the size of the data base to be processed. In this case we were able to reduce the size in close to 93.78%. Originally we had to deal with 166 million elements (i.e 400,000 registers with 415 attributes each); instead we used a simple with only 10.32 million such elements (80,000 records with 129 attributes). The reduced sample, however, performed in a way that made it statistically indistinguishable from the original data. Apart from the benefit resulting from having quicker access to strategic information the use of this methodology yields economic benefits derived from the ability to process a smaller sample (increased speed and capacity for data processing; decreased amount of primary and secondary storage, costs of software and hardware, among others). Considering that the company had important improvements with the application of the results of this investigation, we consider that continuing research is needed and justified, since much work remains to be done if we wish to set a bound on the characteristics of the data which will allow us to generalize the results reported here.

Acknowledgments

Although the determination of the experimental probability distributions was achieved from the application of software designed by the authors, we wish to acknowledge that the graphs shown were obtained with CurveExpert v.1.3 (<http://curveexpert.webhop.biz/>) and clustering was performed with IBM® Intelligent Miner v.6.1

References

1. Palpanas, T.: Knowledge Discovery in Data Warehouses. ACM SIGMOD record. 29(3), 88–100 (2000)
2. Jain, K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Computing Surveys 31(3), 264–323 (1999)
3. Berkhin, P.: Survey of Clustering Data Mining Techniques. Accrue Software (2002)
4. Kleinberg, J., Papadimitriou, C., Raghavan, P.: Segmentation Problems. Journal of the ACM 51(2), 263–280 (2004)
5. Guha, S., Rastogi, R., Shim, K.: CURE: An efficient clustering algorithm for Large Databases. In: ACM SIGMOD Proceedings, pp. 73–84. ACM Press, New York (1998)

6. Peter, W., Chiochetti, J., Giardina, C.: New unsupervised clustering algorithm for large datasets. In: ACM SIGKDD Proceedings, pp. 643–648. ACM Press, New York (2003)
7. Raymond, T.N., Jiawei, H.: Efficient and Effective Clustering Methods for Spatial Data Mining. 20th International Conference on Very Large Data Bases, pp. 144–155 (1994)
8. Cheng, D., Kannan, R., Vempala, S., Wang, G.: A Divide-and-Merge Methodology for Clustering. In: ACM SIGMOD Proceedings, pp. 196–205. ACM Press, New York (2005)
9. Jagadish, H.V., Lakshmanan, L.V., Srivastava, D.: Snakes and Sandwiches: Optimal Clustering Strategies for a Data Warehouse. In: ACM SIGMOD Proceedings, pp. 37–48. ACM Press, New York (1999)
10. Palmer, C.R., Faloutsos, C.: Density Biased Sampling: An Improved Method for Data Mining and Clustering. In: ACM SIGMOD Record, pp. 82–92. ACM Press, New York (2000)
11. Liu, H., Motoda, H.: On Issues of Instance Selection. *Data Mining and Knowledge Discovery*, vol. 6(2), pp. 115–130. Springer, Heidelberg (2002)
12. Zhu, X., Wu, X.: Scalable Representative Instance Selection and Ranking. In: Proceedings of the 18th IEEE international conference on pattern recognition, pp. 352–355. IEEE Computer Society Press, Los Alamitos (2006)
13. Brighton, H., Mellish, C.: Advances in Instance Selection for Instance-Based Learning Algorithms. *Data Mining and Knowledge Discovery* 6, 153–172 (2002)
14. Vu, K., Hua, K.A., Cheng, H., Lang, S.: A Non-Linear Dimensionality-Reduction Technique for Fast Similarity Search in Large Databases. In: ACM SIGMOD Proceedings, pp. 527–538. ACM Press, New York (2006)
15. Zhang, D., Zhou, Z., Chen, S.: Semi-Supervised Dimensionality Reduction. In: Proceedings of the SIAM International Conference on Data Mining (2007)
16. Fodor, I.K.: A survey of dimension reduction techniques. U.S. Department of Energy, Lawrence Livermore National Laboratory (2002)
17. Hair, J.F., Anderson, R.E., Tatham, R.L., Black, W.C.: *Análisis Multivariante*, 5th edn., pp. 11–15. Pearson Prentice Hall, Madrid (1999)
18. Delmater, R., Hancock, M.: *Data Mining Explained: A Manager's Guide to Customer-Centric Business Intelligence* (Chapter 6). Digital press (2001)
19. Bezdek, J.C.: Cluster Validity with Fuzzy Sets. *Journal of Cybernetics* (3), 58–72 (1974)

Combining Traditional and Neural-Based Techniques for Ink Feed Control in a Newspaper Printing Press

Cristofer Englund¹ and Antanas Verikas^{1,2}

¹ Intelligent Systems Laboratory, Halmstad University, Box 823,
S-301 18 Halmstad, Sweden
`cristofer.englund@ide.hh.se`

² Department of Applied Electronics, Kaunas University of Technology, Studentu 50,
LT-3031, Kaunas, Lithuania
`antanas.verikas@ide.hh.se`

Abstract. To achieve robust ink feed control an integrating controller and a multiple models-based controller are combined. Experimentally we have shown that the multiple models-based controller operating in the training region is superior to the integrating controller. However, for data originating from outside the multiple models training region, the integrating controller has the advantage. It is, therefore, suggested to combine the two techniques in order to improve robustness of the control system.

1 Introduction

Colour images, as such appearing on a camera display or a computer monitor are composed of a mixture of three primary colours; red (R), green (G) and blue (B). The *RGB* primaries correspond to the three types of colour sensing elements—*cones*—found in the human eye [1]. *RGB* is an additive colour system meaning that the spectra of the light coming from the three primary sources are added to reproduce the spectrum of a certain colour. When the three primaries are mixed in equal portions a grey shade is conceived. The lower the intensity the darker does the colour appear.

A white substrate, for example paper, is usually used in printing. White paper possesses approximately the same reflection coefficient for all wavelengths in the visible spectrum. When the white paper is illuminated, the colour perceived by the observer approximately matches that of the light source. To attain colours during printing, in contrast to the additive system, a subtractive colour system is used where portions of the light are absorbed by the printed ink. The type of ink determines in what part of the spectrum the absorption takes place. The primary colours usually used in four-colour printing are cyan (C), magenta (M), yellow (Y), and black (K), *CMYK*. A *CMY* overprint creates black colour. However, black ink (K) is also used in printing. Due to economical reasons, black ink often replaces the *CMY* overprints. Moreover, black ink is often used to improve the

quality of colour pictures. Since colour images are usually obtained in the *RGB* colour space, while printed using the *CMYK* primaries, printing involves the so called colour separation process, where *RGB* images are transformed into the *CMYK* colour space [2,3].

Usually the printing press operator samples the print manually throughout the job run. The sample is compared to the approved sample print and an effort is made to compensate for colour deviations detected in the print. Each operator performs the adjustments based on the experience gained from working at the press. Typically, the perception of the printed result is very subjective and consequently great variations may appear in the printed result. By using an automatic control system one can eliminate the inconsistent sampling and subjective colour compensations made by the operator and one can expect a number of favourable affects on the printed result, i.e. more uniform print quality through the production. Amongst the advantages of using an automatic control system are the continuous sampling, its swiftness, the consistency in control actions and that it is indefatigable. Operator's time is also set free for the benefit of service and maintenance of the printing press equipment.

There are few successful attempts to automatically control the ink feed in an offset newspaper printing press. In [4], a decision support system is discussed. The print is measured and a knowledge base, build up from observing an experienced operator, is used to help a novice operator to adjust the printing press to compensate for ink density deviations in the print. The decision support system developed in [5] is used in a wall-covering rotogravure printing industry. The system measures a number of characteristics of the print, including colour. If drift is detected in any of the parameters, the system instructs the operator to make adequate adjustments to the process variables. The system developed in [6] for online ink feed control is able to drive the ink density of the print, to the desired target density level. The decision support system developed in [7] has been developed for defect recognition and misprint diagnosis in offset printing. The system is able to recognize defects based on an image sensor, classify the defects into one of 47 categories including color drift and suggests what action the operator should take to eliminate the cause of defect.

In all the aforementioned works, the ink feed control is based on controlling the ink density measured on a solid print area, as that shown on the left of Fig. 1. However, printed pictures are made of dots, see the image on the right of Fig. 1. Since not only the ink density, but also the size of the dots may vary in the printing process, ink density does not provide enough information for controlling the printing process. The amount of ink integrating information on both the ink density and the dot size should be used instead. Therefore, the printed amount of ink estimated in the double grey bar, shown in the image on the right of Fig. 1, is the control variable used in this work. The double grey bar consists of two parts, one part is printed using the black ink and the other part using the cyan, magenta and yellow inks.

We use the technique proposed in [8] to estimate the amount of ink in the double grey bar. To estimate the amount of ink, the *RGB* image recorded from

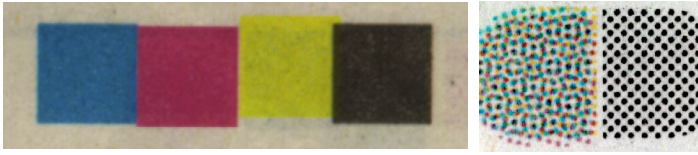


Fig. 1. *Left:* Solid print areas. *Right:* The double grey bar.

the double grey bar by a colour CCD camera is transformed into the $L^* a^* b^*$ counterpart and the average $L^* a^* b^*$ values are calculated for both parts of the double grey bar. The neural networks-based technique [8] then transforms the pair of $L^* a^* b^*$ values into the amount of inks. The neural network is trained using colour patches printed with constant ink density and varying tonal value (the percentage of area covered by the ink). If the ink density used to print a test patch is equal to that kept when printing patches for training the neural network, the printed amount of ink may vary between 0 and 100. If the ink density exceeds the one used to print the training patch, the measured amount of ink may exceed 100 (for an area with 100% ink coverage). In this work, the given amount of ink is the target signal the controller has to maintain.

An approach to automatic data mining and printing press modelling has recently been proposed [9]. Based on this approach we have developed a multiple models-based technique for ink feed control, which has shown good performance in controlling the ink flow in an offset printing press [10]. There are a number of models of different complexity, specialised and general ones, engaged in controlling the printing process. The specialised models are trained on specialised data sets, while the general models are trained on the union of the data sets used to train the specialised models. A committee of specialised models is also incorporated into the set of multiple models. By using the adaptive data mining and modelling approach we provide the multiple models-based controller with up to date models.

Multiple models-based controllers have shown to be efficient in many industrial control applications due to their ability to improve stability and increase the modelling performance [11,12,13]. However, neural networks-based models run into generalisation problems when data outside the training region need to be processed. Such situations are encountered in printing industry, since new unknown jobs may always appear. To cope with the problems we suggest building a hybrid control system consisting of an integrating controller and a multiple models-based controller. In industry applications, integrating controllers are commonly used due to their simplicity and efficiency.

2 Description of Process Variables

The printing press operator samples the print throughout the job run. As colour deviation from the approved sample print is detected the ink flow is changed, increased or decreased, by adjusting the ink keys. The ink keys are situated at

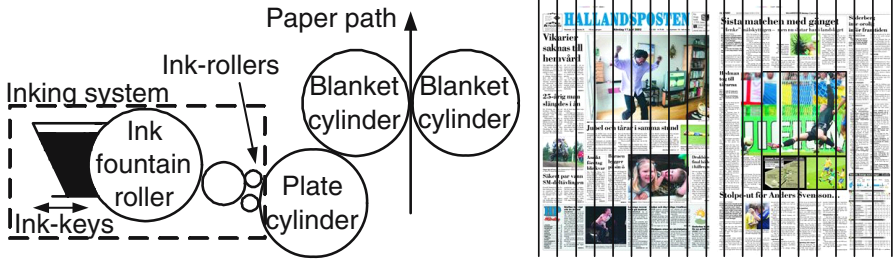


Fig. 2. *Left:* A schematic illustration of the inking system. *Right:* An illustration of how the ink zones subdivide the paper fold.

the bottom of the ink tray, Fig. 2 (left). At the press at hand there are 36 ink keys for each colour and side of the web. The ink key adjusts the ink feed in an approximately 4 cm wide zone (ink zone), see Fig. 2 (right).

In the present work, to create a multiple models-based controller for adjusting the ink key opening, we utilise models of the printing process, build from historical process data. The data collected in one ink zone are called specialised data and hence, used to train the specialised models. The union of all the specialised data is used to train the general models. Both inverse models, where the ink key opening value constitutes the model output and direct models, where the printed amount of ink constitutes the output, are built. The process parameters used to model one ink (C , M , Y , or K) are given below. Depending on the modelling task, inverse or direct, different combinations of these parameters are utilised.

x_1 — printing speed in copies per hour.

x_2 — ink fountain roller speed.

x_3 — ink temperature. The temperature of the ink in the ink tray. The temperature affects the viscosity of the ink. The higher the temperature the lower the viscosity—the easier does the ink flow through the inking system.

$x_{4,5,6}$ — estimated ink demand for the current, adjacent to the left, and to the right ink zone, respectively. The ink demand equals to the percentage of area covered by ink in the corresponding ink zone.

$x_{7,8,9}$ — ink key opening for the current, adjacent to the left, and to the right ink zone, respectively—is the signal controlling the amount of ink dispersed on the paper.

x_{10} — amount of ink of a specific colour estimated from the double grey bar.

In the direct modelling, $x_{10}(t+1)$ is the model output. However, for inverse modelling, where the modelling task is to predict the ink key opening, the $x_{10}(t+1)$ value is used as an input parameter, while the parameter $x_7(t+1)$ constitutes the model output. The variables x_7 and x_{10} are used from both the current time step (t) and the next ($t+1$). Experimental studies have shown that no further performance gain is achieved by exploiting more previous time steps e.g. ($t-1$) or ($t-2$). The variables $x_{4,5,6}$, describe the ink demand in the current zone (x_4)

and the two adjacent zones (x_5, x_6) . Since ink flows between adjacent zones in the printing press, the variables $x_{4,5,6}$ are replaced by their mean $\overline{x_{4,5,6}}$, in the models. For simplicity we denote the variables incorporated in the direct and inverse model as:

$$\mathbf{v}^d = [x_1(t), x_2(t), x_3(t), \overline{x_{4,5,6}(t)}, x_7(t+1), x_7(t), x_8(t), x_9(t), x_{10}(t)] \quad (1)$$

$$\mathbf{v}^i = [x_1(t), x_2(t), x_3(t), \overline{x_{4,5,6}(t)}, x_7(t), x_8(t), x_9(t), x_{10}(t+1), x_{10}(t)] \quad (2)$$

It should be noted that these variables are used to train the models. When the models are used for control the variable $x_7(t+1)$ is replaced by the output of the inverse model $u(t+1)$ and $x_{10}(t+1)$ is replaced by the desired amount of ink y^{des} . Note also that the ink key opening value varies in the range $[0,100]$. To obtain data necessary for the modelling, a web offset newspaper printing press was equipped with an online press monitoring system. A detailed description of the monitoring system can be found in [10].

3 Methods

3.1 Printing Process Modelling

Due to wear of the printing press, the process can be classified as slowly time-varying. In addition, depending on a printing job, the time the process stays in a predefined part of the input variable space may vary significantly, from minutes to several days. If the process starts to operate in a new region of the input variable space, different from the training region, the model performance may deteriorate significantly. To handle such situations, we have recently proposed an adaptive data mining and modelling approach [9]. The data mining tool monitors the process data and keeps an up to date data set of a reasonable size characterising the process. The adaptive modelling is aiming at building models of optimal complexity. Starting with a linear model, a number of nonlinear models of increasing complexity (MLP with an increasing number of hidden units) are built. Then a model with the lowest generalisation error is selected for modelling the process. During the process run, the need to update the models is automatically detected and the models are retrained. In this work, we use this technique to create and update the process models.

Four types of models are used in this work for modelling the printing process.

- A model specific for each ink key/zone. These models are called specialised, since they have specific knowledge about a certain ink key/zone. Each specialised model is trained using data from a specific ink zone.
- A committee of specialised models. Specialised models implementing similar functions are aggregated into a committee. In [14] we developed an approach for building committees of models where both the number of members and the aggregation weights of the members are data dependent. We use this approach to create committees of models.

- A nonlinear general model that is built using the data from all the ink-zones. The general model is built using more data than the specialised one and therefore it generalises better than the specialised models.
- A linear general model built using data from all the ink zones.

The specialised models and committees of the models provide the highest modelling accuracy. However, due to the limited training data set used, the models may run into generalisation problems. In such situations, general models are used instead, which are built using much more data points than the specialised ones. Since the complexity of the models is determined automatically the general model may be linear or nonlinear. If a nonlinear general model is automatically selected, a linear general model is also built. The linear general model exhibits the lowest modelling accuracy, however the best generalisation ability.

4 Ink Key Control

The data acquisition system is not only capable of reading the status of the printing press control system but also sending control signals to the press. The process controller was implemented in a closed loop control system where the signals to and from the controller are sent via the data acquisition system. The sampling time is approximately 100 seconds i.e. the time needed for the monitoring system to traverse the camera once over the paper web to take an image of each of the 36 double grey bars and return to the initial position.

Model-based control systems are common in industry because process models have the ability to mimic both the direct and inverse behavior of the process. Multiple models-based controllers have shown to be efficient in different industrial control applications due to their ability to improve stability and increase the modelling performance [11,12,13]. A detailed description of the multiple models-based design we have developed for ink feed control can be found in [10]. Here we provide only a brief summary of the technique.

4.1 Multiple Models-Based Controller Design

Fig. 3 illustrates the multiple models-based configuration, where the denotation IM stands for inverse model and DM means direct model. Models incorporated in the control configuration are:

- Sing*—a single specialised model.
- Com*—a committee of specialised models.
- NLGen*—a single general nonlinear model.
- LGen*—a single general linear model.

The control configuration functions as follows. The control signal $u(t+1)$ is given by the output of one of the inverse models. We assume that the inverse model output is normally distributed with the mean given by the model output and the standard deviation σ . A large standard deviation of the predicted control

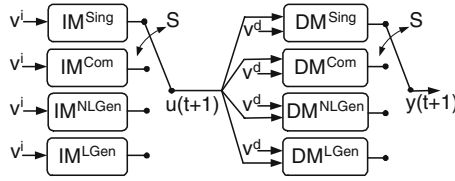


Fig. 3. The multiple models-based control configuration

signal indicates model uncertainty. By sampling from the distribution of the inverse model output, as suggested in [15], we produce a set of control samples $U(t + 1) = [u_1(t + 1), u_2(t + 1), \dots, u_D(t + 1)]$ that are evaluated using the direct model, see Fig. 3. The number of samples D is determined by the model standard deviation σ . The larger the σ the more samples are generated.

The output of the inverse model $u(t + 1)$ and the direct model $y(t + 1)$ are given by

$$u(t + 1) = f^i(\mathbf{v}^i; \boldsymbol{\theta}^i) \tag{3}$$

$$y(t + 1) = f^d(\mathbf{v}^d; \boldsymbol{\theta}^d) \tag{4}$$

where $\boldsymbol{\theta}$ is the model parameter vector and the functions f are either linear or nonlinear.

The control signals $u_{i1}(t + 1), u_{i2}(t + 1), \dots, u_{iD}(t + 1)$ generated by each of the inverse models ($i = 1, \dots, 4$) are used to calculate the outputs $y_{11}(t + 1), y_{21}(t + 1), \dots, y_{41}(t + 1), \dots, y_{4D}(t + 1)$ of the direct models. The output $y_{ij}(t + 1)$ is given by

$$y_{ij}(t + 1) = f^d(\mathbf{v}_{ij}^d; \boldsymbol{\theta}^d) \tag{5}$$

where, $i = 1, \dots, 4$ refers to a model. The model selected is that minimising the error e_{ij} , the difference between the output of the direct model $y_{ij}(t + 1)$ and the target (the desired amount of ink) y^{des} : $e_{ij} = ||y_{ij}(t + 1) - y^{des}||$. Having all e_{ij} s, the indices p, q of the control signal $u_{pq}(t + 1)$ sent to the press are found as follows:

$$p, q = \arg \min_{i,j} e_{ij} \tag{6}$$

The control signal selected is denoted $u^{mm}(t + 1)$. If for a given \mathbf{v} , $e_{pq} > \beta$ and $p \neq 3$, the linear general model is used to avoid using the nonlinear model with a large prediction error.

4.2 Robust Ink Feed Control

Fig. 4 illustrates the case, where the neural networks-based controller runs into generalisation problems. The left graph shows the ink key control signal (above) and the measured amount of ink along with the target amount of ink indicated

by the solid line (below). Initially the multiple models-based controller runs the process. At sample no. 7 and 9 the target amount of ink is changed. Accordingly, the multiple models-based controller is adjusting the ink key opening to obtain the desired amount of ink. As it can be seen, the ink key adjustments do not bring the process output to the desired level. At sample no. 21 and 22 the control action from the multiple models-based controller is manually overridden and the desired target level is reached. Fig. 4 (right) explains the origin of the problem. The right graph of Fig. 4 shows the training data (*) and the data from the current job (Δ and \square) projected onto the first two principal components of the training data. We clearly see that the data indicated by the squares are well separated from the training data. It is obvious that to successfully use the multiple models-based controller the models need to be retrained. However, to retrain the models, training data are to be collected. We suggest using an integrating controller during this period of time. Though with lower accuracy, the integrating controller can handle the process temporary.

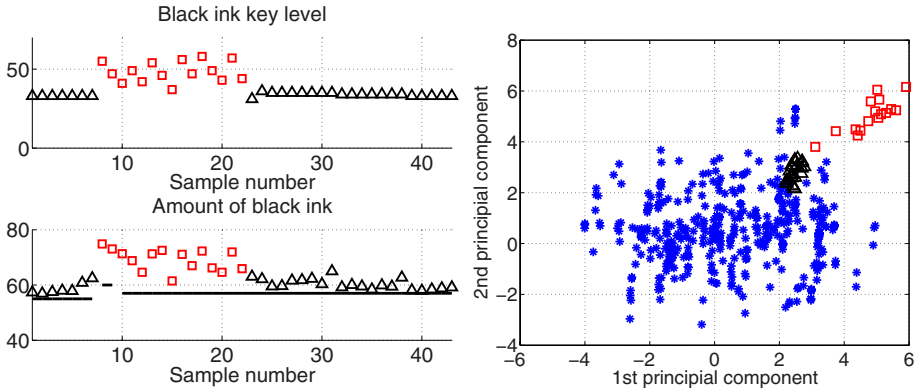


Fig. 4. *Left (top):* Ink key opening and (*bottom*): the measured and the target (solid line) amount of ink. *Right:* The data projected onto the space spanned by the first two principal components.

We use the difference between the predicted amount of ink $y(t - 1)$ and the measured amount of ink at time t , $y^{mes}(t)$ to detect the situations. The schematic illustration of the robust ink feed controller is shown in Fig. 5. The control signal $u(t + 1)$ is given by:

$$u(t + 1) = \begin{cases} u^{ic}(t + 1) & \text{if } (y^{mes}(t) - y(t)) > \xi \\ u^{mm}(t + 1) & \text{otherwise} \end{cases} \quad (7)$$

where $u^{mm}(t + 1)$ is the ink key opening predicted by the multiple models, $u^{ic}(t + 1)$ is the ink key opening predicted by the integrating controller, $y(t)$ is the amount of ink predicted by the multiple models at $t - 1$, and $y^{mes}(t)$ is the measured amount of ink at the current time step t . By using this approach

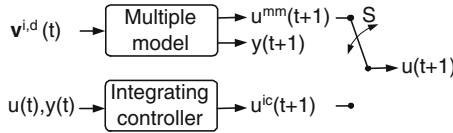


Fig. 5. The proposed control configuration

the process is controlled, either by the integrating or multiple models-based controller.

4.3 Integrating Controller Design

The control signal generated by the integrating controller is estimated as

$$u^{ic}(t+1) = u(t) + K(y^{des} - y^{mes}(t)) \quad (8)$$

where $u(t)$ is the ink key opening at the time step t , K is the integrating factor, and y^{des} is the desired amount of ink.

5 Experimental Investigations

The experiments have been made during normal production at the offset printing-shop. The experiments were conducted to investigate three matters:

1. To find the appropriate value of the parameter K for the integrating controller.
2. To compare the integrating and the multiple models-based controllers.
3. To demonstrate the benefit of the proposed control configuration.

5.1 Selecting the Parameter K

To find the appropriate K value, the parameter was varied between 0.2 and 2.5. In Fig. 6 we present three examples of control and output signals for different K parameter values. The top graph shows the control signal, the lower graph shows the measured and the desired (the solid line) amount of ink. The desired amount of ink is constant during the experiment. The controller starts running the process at sample 4. As it can be seen, the larger the K , the larger is the control action.

It was found that $K=0.7$ is a good choice since at this value, on average, the controller was reasonably fast and not too sensitive to noise. As it can be seen in Fig. 6 at $K = 0.2$ the rise time is very long. At $K=1.4$ both the control signal and the output signal are rather noisy. The standard deviation of the output signal (noise level) is 3.3, 2.5 and 2.3 for $K=0.2$, 0.7 and 1.4, respectively.

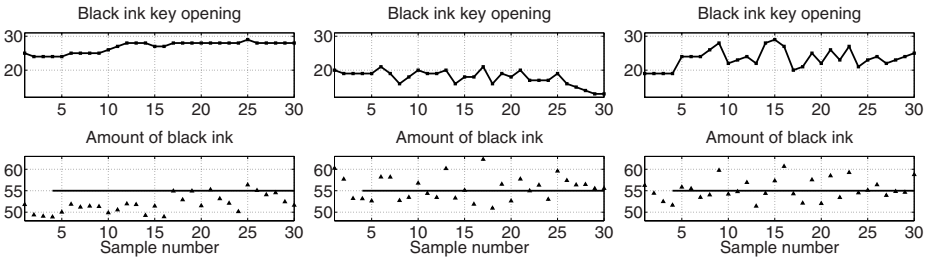


Fig. 6. The control signal (top) and the measured along with the desired (solid line) amount of ink for different K values (bottom). $K = 0.2, 0.7$ and 1.4 for the left, middle, and the right graph, respectively.

5.2 Comparison of the Controllers

To make the comparison feasible, we use the controllers in the same ink zone for the same printing job. We begin by saving the initial settings for the press and start the experiment using one of the controllers. Then, we restore the settings of the printing press and continue the same experiment using the other controller. Two issues are studied, the rise time and the sensitivity to noise.

Rise Time. A short rise time is desirable to reduce the paper waste. In Fig. 7, we present the response of the controllers operating on the same ink key for two different colours. For each colour, the left graphs show the results from the integrating controller, whereas the right graphs present the results from the multiple models-based controller. The top graphs show the ink key control signal and the bottom graphs present the measured and the target (solid line) amount of inks. The controller is used from sample 3 (where the solid line appears). In the figures, ID stands for ink demand.

As it can be seen, the integrating controller requires more samples to drive the output to the desired target level. The multiple models-based controller exhibits a shorter rise time than the integrating controller.

Fig. 8 presents two more control examples. The results presented are for the case where the target amount of ink is less than the initial printed amount of ink. Again, for both examples, the multiple models-based controller drives the amount of ink to the desired level faster than the integrating controller.

Noise in the Control and Output Signals. Our previous studies have shown that, on average, the noise level in the measured amount of ink is approximately equal to 2 units [9]. The examples presented show that the integrating controller does not produce as stable output as the multiple models-based controller does. On average, the noise level for the integrating controller was larger than for the multiple models-based controller. Table 1 summarises the standard deviation $\sqrt{\frac{1}{N-1} \sum (y^{mes} - y^{des})^2}$, of the output signal, for the examples presented in Fig. 7 and Fig. 8 and for the long time experiments carried out during normal

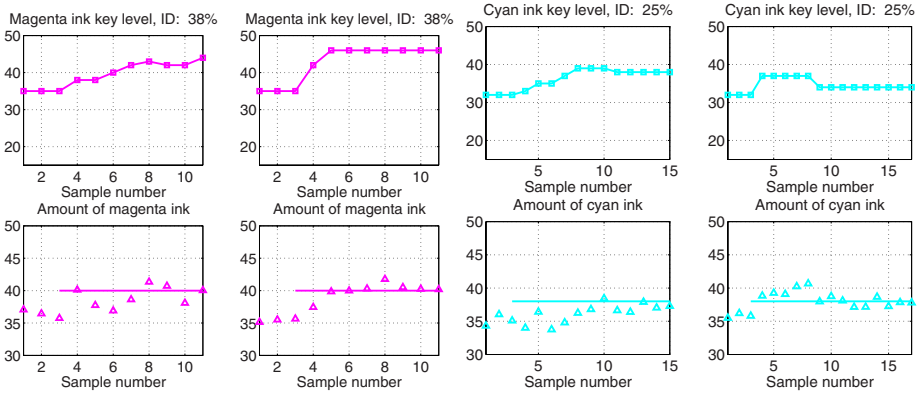


Fig. 7. Results from the integrating controller (first and third columns) and the multiple models-based controller

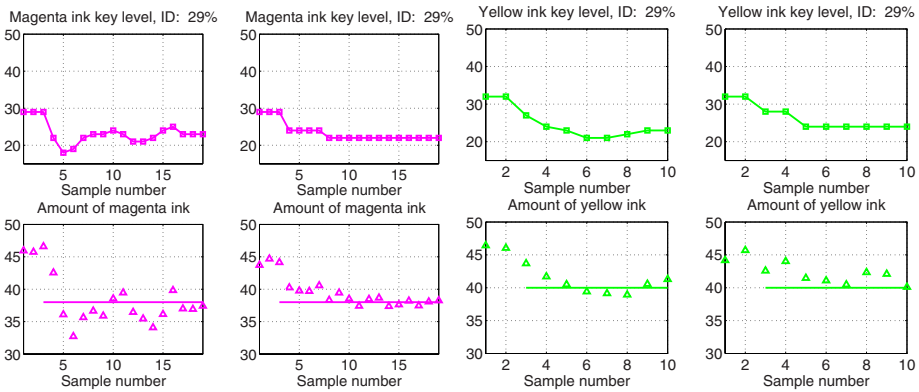


Fig. 8. Results from the integrating controller (first and third columns) and the multiple models-based controller

production. Observe that the target amount of ink was constant. The long time experiments lasted for 3 hours.

5.3 Robust Ink Feed Control

The printing process may start to operate in a new region of the input variable space, different from the training region, as it was discussed earlier and illustrated in Fig. 4. Fig. 9 presents an example illustrating the benefit of the approach proposed in such situations. The top left graph in Fig. 9 shows the ink key control signal. The target (solid line) and the measured amount of inks are shown in the middle left graph. The prediction error and the threshold of the error are presented in the bottom left graph. We distinguish three regions in the control sequence. At the beginning, the multiple models-based controller runs

Table 1. The standard deviation of the measured amount of ink for the experiments illustrated in Fig. 7, Fig. 8, and for the long time measurements (LT). IC stands for integrating controller, MM for multiple models-based controller, and C, M, Y, K for cyan, magenta, yellow, and black.

Controller	Fig. 7(M)	Fig. 7(C)	Fig. 8(M)	Fig. 8(Y)	LT(C)	LT(M)	LT(Y)	LT(K)
IC	1.68	3.79	3.70	2.91	5.8	4.5	4.3	5.7
MM	1.12	1.97	1.65	1.94	2.5	2.7	2.4	3.4

the process. At the point where the difference between the predicted and the measured amount of ink exceeds the threshold, $\xi = 6$, the integrating controller takes over the control, samples indicated by diamonds (\diamond). The integrating controller brings the process to the target amount of ink and the prediction error of the model is low again. The right graph in Fig. 9 shows the input data projected onto the first two principal components. As it can be seen the data resulting in high prediction error appear at the edge of the main bulk of the training data (shown as stars *). This explains why the multiple models-based controller has problems with these data points.

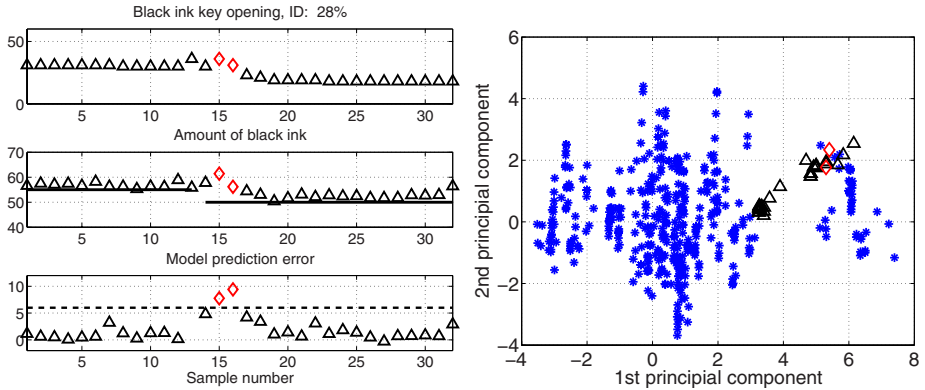


Fig. 9. *Left above:* The ink key opening. *Left middle:* The measured and the target (solid line) amount of ink. *Left below:* The error of the predicted amount of ink and the threshold of the error (dashed line). *Right:* The data projected onto the first two principal components.

6 Conclusions

A technique for robust ink feed control in an offset lithographic printing press has been presented in this paper. The technique combines a traditional integrating controller and a neural networks-based (multiple modes-based) controller. We have shown that the multiple models-based controller is superior to the integrating controller by both lower rise time and lower noise in the output signal. However, as the process starts operating in a new region of the input space,

the multiple models may run into generalisation problems. Such situations are automatically detected and the integrating controller temporary takes over the process control. We have shown experimentally that the proposed technique is able to automatically control the ink feed in the newspaper printing press according to the target amount of ink.

In future work long term experiments will be carried out for a wide variety of printing jobs where the performance of the system under the influence of disturbances such as reel changes, temperature and printing speed changes, etc will be investigated.

Acknowledgements

We gratefully acknowledge the financial support from the Knowledge Foundation Sweden and Holmen Paper, StoraEnso, and VTAB groups Sweden.

References

1. Sharma, G., Trussell, H.J.: Digital color imaging. *IEEE Transactions on Image Processing* 6, 901–932 (1997)
2. Balasubramanian, R.: Optimization of the spectral Neugebauer model for printer characterization. *Journal of Electronic Imaging* 8, 156–166 (1999)
3. Pappas, T.: Model-based halftoning of color images. *IEEE Transactions on Image processing* 6, 1014–1024 (1997)
4. Almutawa, S., Moon, Y.B.: The development of a connectionist expert system for compensation of color deviation in offset lithographic printing. *AI in Engineering* 13, 427–434 (1999)
5. Brown, N., Jackson, M., Bamforth, P.: Machine vision in conjunction with a knowledge-based system for semi-automatic control of a gravure printing process. In: *Proceedings of the IMECH E Part I Journal of Systems & Control Engineering*, vol. 218, pp. 583–593. Professional Engineering Publishing (2004)
6. Pope, B., Sweeney, J.: Performance of an on-line closed-loop color control system. In: *TAGA 2000 Proceedings*, pp. 417–431 (2000)
7. Perner, P.: Knowledge-based image inspection system for automatic defect recognition, classification and process diagnosis. *Mashine Vision and Applications* 7, 135–147 (1994)
8. Verikas, A., Malmqvist, K., Bergman, L.: Neural networks based colour measuring for process monitoring and control in multicoloured newspaper printing. *Neural Computing & Applications* 9, 227–242 (2000)
9. Englund, C., Verikas, A.: A SOM based data mining strategy for adaptive modelling of an offset lithographic printing process. *Engineering Applications of Artificial Intelligence* 20, 391–400 (2007)
10. Englund, C., Verikas, A.: Ink flow control by multiple models in an offset lithographic printing process. *Computers & Industrial Engineering* (in review) (2006)
11. Chen, L., Narendra, K.S.: Nonlinear adaptive control using neural networks and multiple models. *Automatica* 37, 1245–1255 (2001)
12. Ravindranathan, M., Leitch, R.: Model switching in intelligent control systems. *AI in Engineering* 13, 175–187 (1999)

13. Yu, W.: Multiple recurrent neural networks for stable adaptive control. *Neurocomputing* 70, 430–444 (2006)
14. Englund, C., Verikas, A.: A SOM based model combination strategy. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) *ISNN 2005*. LNCS, vol. 3496(Part 1), pp. 461–466. Springer, Heidelberg (2005)
15. Herzallah, R., Lowe, D.: A mixture density network approach to modelling and exploiting uncertainty in nonlinear control problems. *Engineering Applications of Artificial Intelligence* 17, 145–158 (2004)

Active Learning Strategies: A Case Study for Detection of Emotions in Speech

Alexis Bondu, Vincent Lemaire, and Barbara Poulain

R&D France Telecom,
TECH/EASY/TSI
2 avenue Pierre Marzin 22300 Lannion

Abstract. Machine learning indicates methods and algorithms which allow a model to learn a behavior thanks to examples. Active learning gathers methods which select examples used to build a training set for the predictive model. All the strategies aim to use the less examples as possible and to select the most informative examples. After having formalized the active learning problem and after having located it in the literature, this article synthesizes in the first part the main approaches of active learning. Taking into account emotions in Human-machine interactions can be helpful for intelligent systems designing. The main difficulty, for the conception of calls center's automatic shunting system, is the cost of data labeling. The last section of this paper propose to reduce this cost thanks to two active learning strategies. The study is based on real data resulting from the use of a vocal stock exchange server.

1 Introduction

Active learning methods come from a parallel between active educational methods and learning theory. The learner is from now a statistical model and not a student. The interactions between the student and the teacher correspond to the opportunity (possibility) to the model to interact with a human expert. The examples are situations used by the model to generate knowledge on the problem.

Active learning methods allow the model to interact with its environment by selecting the more “informative” situations. The purpose is to train a model which uses as little as possible examples. The elaboration of the training set is done in interaction with a human expert to maximize progress of the model. The model must be able to detect the more informative examples for its learning and to ask to the expert: “what should be done in these situations”.

The purpose of this paper is to present two main active learning approaches found in the state of the art. These approaches are presented in a generic way without considered a kind of model (the one which learns using examples delivered by the expert after every of its requests). Others approaches exist but they are not presented in this paper although references are given for the reader who would be interested in.

The next section of this paper introduce the topic, formalize active learning in a generic way and establish mathematical notations used. The aim of this section is to place active learning among others statistical learning methods (supervised, unsupervised...). The fourth section presents in details two main active learning approaches. These two strategies are then used in the fifth section on a real problem. Finally the last section is a discussion on question open in this paper.

2 Active Learning

2.1 General Remarks

The objective of statistical learning (unsupervised, semi-supervised, supervised^[1]) is to “inculcate” a behavior to a model using observations (examples) and a learning algorithm. The observations are points of view on the problem to be resolved and constitute the learning data. At the end of the training stage the model has to generalize its learning to unseen situations in a “reasonable” way.

For example let’s imagine a model which try to detect “happy” and “unhappy” people from passport photo. If the model realizes good predictions for unseen people during its training stage then the model correctly generalize.

Characteristics of used data change depending on the learning mode. Un-supervised learning is a method of machine learning where a model is fit to observations. It is distinguished from supervised learning by the fact that there are no a priori outputs on data. The learner has to discover itself correlations between examples which are shown to it. In case of the example above (“happy” / “unhappy” people), the model is trained using passport photos deprive of label and has no indication on what we try to make it learn. Among unsupervised learning methods one finds clustering methods [2] and association rules methods [3].

Semi-supervised learning [4] is a class of techniques that makes use of both labeled and unlabeled data for training; typically a small amount of labeled data and a large amount of unlabeled data. Among possible utilization of this learning mode, we could distinguish (i) semi-supervised clustering which tries to group similar instances but using information given by the small amount of labeled data [5] and (ii) semi-supervised classification [6] which is based first on labeled data to elaborate a first model and then unlabeled data to improve the model.

Supervised learning is a machine learning technique for creating a function from training data. The training data consist of pairs of input objects (typically vectors), and desired outputs. The output of the function can be a continuous value (called regression), or can predict a class label of the input object (called classification). The task of the supervised learner is to predict the value of the function for any valid input object after having seen a number of training examples (i.e. pairs of input and target output). In case of the illustrative example above, examples would be passport photos associated to labels “happy” or “unhappy”.

¹ Reinforcement learning is not presented here, reader interested could read [1].

At last, active learning, as the name suggests, is a type of learner which is less passive than the others described above. This strategy allows the model to construct its own training set in interaction with a human expert. The learning starts with few desired outputs (class labels for classification or continuous value for regression). Then, the model selects examples (without desired outputs) that it considers the more informative and asks to the human expert their desired outputs. In case of the illustrative example, the model asks class labels of passport photos presented to the human expert. In this paper, we restrict active learning to classification but it is obvious that our presentation of active learning strategies can be transpose for regression.

Active learning is different from all others learning methods because it interacts with its environment; the examples are not randomly chosen. Active learning strategies allow the model to learn faster (the learner rich the best performances using less data) considering first the more informative examples. This approach is more specifically attractive when data are expensive to obtain or to label.

2.2 Two Possible Scenarios

The distinction between raw data and data descriptors (which are associated) is important. In the illustrative example above, the raw data are passport photos and the data descriptors are attributes describing the photos (pixel, luminosity, contrasts, etc.). The model makes the prediction of the class “happy” or “unhappy” to every vector of descriptors. The elaboration of descriptors from raw data is not always bijective; sometimes it is impossible to compose raw data using list of descriptors. Adaptive sampling and selective sampling, which are the two main scenarios to set active learning [7], use respectively “data descriptors” and “raw data”.

In the case of **adaptive sampling** [8] the model requires of expert labels corresponding to vectors of descriptors. The model is not restricted and can explore all the space of variations of the descriptors, searching area to be sampled more finely. Adaptive sampling can pose problem in its implementation when it is difficult to know if the vectors of descriptors (generated by the model) have a meaning with respect to the initial problem. Let us suppose that the model requires the label associated with the vector $[10, 4, 5, \dots, 12]$. Does this vector correspond to a set of descriptors which represent a passport photo, a human face photo, a flower or something else?

In the case of **selective sampling** [9], the model observes only one restricted part of the universe materialized by training examples stripped of label. Consequently, the input vectors selected by the model always correspond to a raw data. The image of a “*bag*” of instances for which the model can ask labels associated (to the examples in the bag) is usually used. The model requires the label associated with the vector $[10, 4, 5, \dots, 12]$ which corresponds to a passport photo.

Emotion detection is a problem where it is easy to obtain a great number of unlabeled examples and for which labeling is expensive. Therefore, from now the point of view of selective sampling is only considered. In practice, the choice of

selective or adaptive sampling depends primarily on the applicability where the model is authorized, or not, “to generate” new examples.

2.3 Notations

$\mathcal{M} \in \mathbb{M}$ is the predictive model which is trained thanks to an algorithm \mathcal{L} . $\mathbb{X} \subseteq \mathbb{R}^n$ represents all the possible input examples of the model and $x \in \mathbb{X}$ is a particular examples. \mathbb{Y} is the set of the possible outputs (answers) of the model; $y \in \mathbb{Y}$ a class label² related (associated) to $x \in \mathbb{X}$.

During its training, the model observes (see Figure 1) only one part $\Phi \subseteq \mathbb{X}$ of the universe. The set of examples is limited and the labels associated to these examples are not necessarily known. The set of examples where the labels are known (at a step of the training algorithm) is called L_x and the set of examples where the labels are unknown is called U_x with $\Phi \equiv U_x \cup L_x$ and $U_x \cap L_x \equiv \emptyset$.

The concept which is learned can be seen as a function, $f : \mathbb{X} \rightarrow \mathbb{Y}$, with $f(x_1)$ is the desired answer of the model for the example x_1 and $\hat{f} : \mathbb{X} \rightarrow \mathbb{Y}$ the obtained answer of the model; an estimation of the concept. The elements of L_x and the associated labels constitute a training set T . The training examples are pairs of input vectors and desired labels such as $(x, f(x)) : \forall x \in L_x, \exists!(x, f(x)) \in T$.

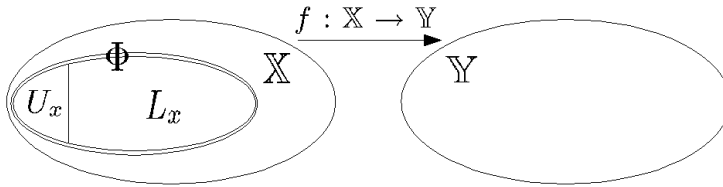


Fig. 1. Notations

3 Active Learning Methods

3.1 Introduction

The problem of selective sampling was posed formally by Muslea [10] (see Algorithm 1). It uses an utility function, $Utility(u, \mathcal{M})$, which estimates the utility of an example u for the training of the model \mathcal{M} . Thanks to this function, the model presents to the expert examples for which it hopes the greatest improvement of its performances.

The Algorithm 1 is generic insofar as only the function $Utility(u, \mathcal{M})$ must be modified to express a particular active learning strategy. How to measure the interest of an example will be discuss now.

² The word “label” is used here for a discrete value in classification problems or a continuous value in regression problems.

Considering:

- \mathcal{M} a predictive model provided with a training algorithm \mathcal{L}
- U_x et L_x the sets of examples respectively not labeled and labeled
- n the desired number of training examples
- T the training set with $\|T\| < n$
- $\mathcal{U} : \mathbb{X} \times \mathbb{M} \rightarrow \mathbb{R}$ the function which estimates the utility of an example for the training of the model

Repeat

- (A) Train the model \mathcal{M} thanks to \mathcal{L} and T (and possibly U_x).
- (B) Look the example such as $q = \operatorname{argmax}_{u \in U_x} \mathcal{U}(u, \mathcal{M})$
- (C) Withdraw q of U_x and ask the label $f(q)$ to the expert.
- (D) Add q to L_x and add $(q, f(q))$ to T

until $\|T\| < n$

Algorithm 1. Selective sampling, Muslea 2002

3.2 Uncertainty Sampling

Uncertainty sampling is an active learning strategy [11,12] which is based on confidence that the model has in its predictions. The model used must be able to estimate the reliability of its answers, to provide the probabilities, y_j , to observe each class (j) for an examples u . Thus the model can make a prediction choosing the most probable class for u . The choice of new examples to be labeled proceeds in two steps:

- the model available at the iteration t is used to predict the labels of the unlabeled examples;
- examples with the more uncertain prediction are selected.

The uncertainty of a prediction can also be defined using a threshold of decision. For example (see Figure 2) if the model gives answers between 0 and 1 a threshold is defined to take a decision and say which examples will be classified 0 and those which will be classified 1. The closer an answer of the model is to the threshold of decision, the more uncertain is the decision.

This first approach has the advantage to be intuitive, easy to implement and fast. The uncertainty sampling shows its limits however when the problem to be solved is not separable by the model. Indeed, this strategy will tend to select the examples to be labeled in mixture zones, where there is nothing any more to learn.

3.3 Risk Reduction

The purpose of this approach is to reduce the generalization error, $E(\mathcal{M})$, of the model [13]. It chooses examples to be labeled so as to minimize this error. In

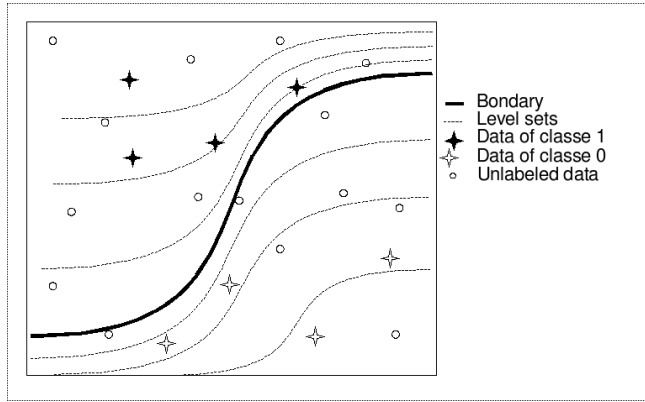


Fig. 2. A binary classification problem: the boundary plotted represents the threshold of decision. Level sets are plotted too and their distances from the boundary lines indicate the uncertainty. Unlabeled data close to the boundary line are the more uncertain and will be selected to be labeled by the expert.

practice this error cannot be calculated because the distribution of the examples, \mathbb{X} , is unknown. However it can be write, at an iteration t , using a loss function ($\mathcal{L}oss(\mathcal{M}^t, x)$) which evaluates the error of the model for a given example $x \in \mathbb{X}$ such as:

$$E(\mathcal{M}^t) = \int_{\mathbb{X}} \mathcal{L}oss(\mathcal{M}^t, x)P(x)dx$$

The same model at the next iteration $t + 1$ is defined as: $\mathcal{M}_{(x^\diamond, y^\diamond)}^{t+1}$. This model takes into account a new training example: (x^\diamond, y^\diamond) . For real problems, the output of the model y^\diamond is unknown since x^\diamond is a not labeled data. To estimate the generalization error at $t+1$, all the possibilities of the \mathbb{Y} set have to be considered and to be balanced using their probability to be observed. The generalization error expected is therefore:

$$E(\mathcal{M}_{x^\diamond}^{t+1}) = \int_{\mathbb{X}} \int_{\mathbb{Y}} P(y|x^\diamond) \mathcal{L}oss(\mathcal{M}_{(x^\diamond, y)}^{t+1}, x)P(x)dx dy$$

This strategies selects the example q which minimizes $E(\mathcal{M}_{x^\diamond}^{t+1})$. Once labeled, this example is incorporated to the training set. Step by step this procedure tries to elaborate an optimal training set.

Nicholas Roy [9] shows how to bring this strategy into play since all the elements of \mathbb{X} are not known. He uses an uniform prior for $P(x)$ which gives :

$$\widehat{E}(\mathcal{M}^t) = \frac{1}{\|L_x\|} \sum_{i=1}^{\|L_x\|} \mathcal{L}oss(\mathcal{M}^t, x_i)$$

This strategy, where different loss functions can be used, is summarized in the algorithm [2]. The model is, for all examples i , trained several times ($\|\mathbb{Y}\|$

times), to estimate $\widehat{E}(\mathcal{M}_{(x_i, y_j)}^{t+1})$. The example i which minimizes the expected loss function ($\widehat{E}(\mathcal{M}_{(x_i)}^{t+1})$) will be incorporated in the training set.

Considering:

- \mathcal{M} a predictive model provided with a training algorithm \mathcal{L}
- U_x and L_x the sets of examples respectively not labeled and labeled
- n the desired number of training examples
- T the training set with $\|T\| < n$
- \mathbb{Y} the label set which can be given to the examples of U_x
- $\mathcal{L}oss : \mathbb{M} \rightarrow \mathfrak{R}$ the generalization error
- $\mathcal{E}rr : U_x \times \mathbb{M} \rightarrow \mathfrak{R}$ the expected generalization error for the model \mathcal{M} trained with an additional example, $T \cup (x_i, f(x_i))$

Repeat

(A) Train the model \mathcal{M} thanks to \mathcal{L} and T

For all examples $x_i \in U_x$ **do**

For all label $y_j \in \mathbb{Y}$ **do**

 i) Train the model $\mathcal{M}_{i,j}$ thanks to \mathcal{L} and $(T \cup (x_i, y_j))$

 ii) Compute the generalization error $\widehat{E}(\mathcal{M}_{(x_i, y_j)}^{t+1})$

end For

 Compute the generalization error

$$\widehat{E}(\mathcal{M}_{x_i}^{t+1}) = \sum_{y_j \in \mathbb{Y}} \widehat{E}(\mathcal{M}_{(x_i, y_j^*)}^{t+1}) \cdot P(y_j | x_i)$$

end For

(B) Look for the example $q = \operatorname{argmin}_{u \in U_x} \widehat{E}(\mathcal{M}_{x_i}^{t+1})$

(C) Withdraw q of U_x and ask the label $f(q)$ to the expert.

(D) Add q to L_x and add $(q, f(q))$ to T

until $\|T\| < n$

Algorithm 2. Apprentissage actif “optimal”, de Nicholas Roy 2000

An example of use of this strategy is presented in [14] where X. Zhu estimates the generalization error ($E(\mathcal{M})$) using the empirical risk:

$$\widehat{E}(\mathcal{M}) = R(\mathcal{M}) = \sum_{n=1}^N \sum_{y_j \in \mathbb{Y}} \{f(l_n) \neq y_j\} P(y_j | l_n) P(l_n) \text{ with } l_n \in L_x$$

where f is the model which estimates the probability that an example belong to a class (a Parzen window [15] in [14]), $P(y_i | l_n)$ the real probability to observe the class y_i for the example $l_n \in L_x$, the indicating function equal to 1 if $f(l_n) \neq y_i$ and equal to 0 if not. Therefore $R(\mathcal{M})$ is the sum of the probabilities that the model makes a bad decision on the training set (L_x).

Using an uniform prior to estimate $P(l_n)$:

$$\hat{R}(\mathcal{M}) = \frac{1}{N} \sum_{n=1}^N \sum_{y_j \in \mathbb{Y}} \{f(l_n) \neq y_j\} \hat{P}(y_j | l_n)$$

The expected cost for any single example u ($u \in U_x$) added to the training set (for binary classification problem) is then:

$$\hat{R}(\mathcal{M}^{+u}) = \sum_{y_j \in \mathbb{Y}} \hat{P}(y_j | u) \hat{R}(\mathcal{M}^{+(u, y_j)}) \text{ with } u \in U_x$$

3.4 Discussion

Both strategies described above are not the only ones which exist. The reader can see a third main strategy which is based on Query by Committee [16,17] and a fourth one where authors focus on a model approach to active learning in a version-space of concepts [18,19,20].

4 Application of Active Learning to Detection of Emotion in Speech

4.1 Introduction

Thanks to recent techniques of speech processing, many automatic phone call centers appear. These vocal servers are used by customers to carry out various tasks conversing with a machine. Companies aim to improve their customer's satisfaction by redirecting them towards a human operator, in the event of difficulty. The shunting of unsatisfied users amounts detecting the negative emotions in their dialogues with the machine, under the assumption that a problem of dialogue generates a particular emotional state in the subject.

The detection of expressed emotions in speech is generally considered as a supervised learning problem. The detection of emotions is limited to a binary classification since taking into account more classes rises problem of the objectivity of labeling task [21]. The acquisition and the labeling of data are expensive in this framework. Active learning can reduce this cost by labeling only the examples considered to be informative for the model.

4.2 Characterization of Data

This study is based on a previous work [22] which characterizes vocal exchanges, in optimal way, for the classification of expressed emotions in speech. The objective is to control the dialogue between users and a vocal server. More precisely, this study deals the relevance of variables describing data, according to the detection of emotions.

The used data result from an experiment involving 32 users who test a stock exchange service implemented on a vocal server. According to the users point of view, the test consists in managing a virtual wallet of stock options, the goal is to realize the strongest profit. The obtained vocal traces constitute the corpus of this study: 5496 “speech turns” exchanged with the machine. Speech turns are characterized by 200 acoustic variables, describing variations of the sound intensity, variations of voice height, frequency of elocution... etc. Data are also characterized by 8 dialogical variables describing the rank of a speech turn in a given dialogue (a dialogue contains several speech turn), the duration of the dialogue... Each speech turn is manually labeled as carrying positive or negative emotions.

The subset of the most informative variables with respect to the detection of expressed emotions in speech is given thanks to a naive Bayesian selector [23]. At the beginning of this process (the selection of the most informative variables), the set of attributes is empty. The attribute which most improves the predictive quality of the model is then added at each iteration. The algorithm stops when the addition of attributes does not improve any more the quality of the model. Finally, 20 variables were selected to characterize vocal exchanges. In this article, used data result from the same corpus and from this previous study. So, every speech turns is characterized by 20 variables.

4.3 The Choice of the Model

The large range of models able to solve classification problems (and sometimes the great number of parameters useful to use them) may represent difficulties to measure the contribution of a learning strategy. A Parzen window, with a Gaussian kernel [15], is used in experiments below since this predictive model uses a single parameter and is able to work with few examples. The “output” of this model is an estimate of the probability to observe the label y_j conditionally to the instance u :

$$\hat{P}(y_j|u) = \frac{\sum_{n=1}^N \{f(l_n)=y_j\} K(u, l_n)}{\sum_{n=1}^N K(u, l_n)} \quad \text{avec } l_n \in L_x \text{ et } u \in U_x \cup L_x \quad (1)$$

where

$$K(u, l_n) = e^{-\frac{\|u-l_n\|^2}{2\sigma^2}}$$

The optimal value ($\sigma^2=0.24$) of the kernel parameter was found thanks to a cross-validation on the average quadratic error, using the whole of available training data [24]. Thereafter, this value is used to fix the Parzen window parameter. The results obtained by this model (using the whole of training data) are similar with the previous results obtained by a naive Bayesian classifier [22]. Consequently, Parzen windows are considered satisfying and valid for the following active learning procedures. Kernel methods and closer neighbors methods are usually used in classification of expressed emotions in speech [25].

The model must be able to assign a label $\hat{f}(u)$ to an input data u , so a decision threshold noted $\mathcal{T}h(L_x)$ is calculated at each iteration. This threshold minimizes the error of the model³ on the available training set. The label attributed is $\hat{f}(u_n) = 1$ if $\{\hat{P}(y_1|u_n) > \mathcal{T}h(L_x)\}$, else $\hat{f}(u_n) = 0$. Since the single parameter of the Parzen window is fixed, the training stage is reduced to count instances (within the meaning of the Gaussian kernel). The strategies of examples selection, without being influenced by the training of the model, are thus comparable.

4.4 Used Active Learning Strategies

Two Active learning strategies are considered in this paper; the active learning strategy which tries to reduce the generalization error of the model and the strategy consists in selecting the instance for which the prediction of the model is most uncertain have been tested.

For the first strategy the Parzen window estimates $P(y_i|l_n)$. The empirical risk is approximated adopting a uniform a priori on the $P(l_n)$. The purpose is to select the unlabeled instance $u_i \in U_x$ which will minimize the risk of the next iteration. $R(\mathcal{M}^{+u_n})$ the “expected” risk resulting from the labeling of the instance u_n (iteration $t + 1$) is estimated. Available labeled data are used to do this estimation when the assumption $f(u_n) = y_1$ [resp $f(u_n) = y_0$] to estimate $\hat{R}(\mathcal{M}^{+(u_n, y_1)})$ [resp $\hat{R}(\mathcal{M}^{+(u_n, y_0)})$] is done.

For the second strategy the *uncertainty* of a prediction is maximum when the output probability of the model approaches the decision threshold.

Apart from these two active strategies, a “stochastic” approach which uniformly selects the examples according to their probability distribution is considered. This last approach play a role of reference used to measure the contribution of the active strategies.

4.5 Results

The presented results come from several experiments on previous learning strategies. Each experiment has been done five times⁴. At the beginning of the experiments, the training set is only constituted by two examples (one positive and one negative) selected randomly. At each iteration, ten examples are selected to be labeled and added to the training set. The considered classification problem, here, is unbalanced: there are 92% of “positive or neutral” emotions and 8% of “negative” emotions. To observe correctly the classification profits (when adding labeled examples), the model evaluation is done using the area under ROC curve (AUC) on the test set⁵. A ROC curve is calculated from the detection rate of

³ The used error measurement is the “Balanced Error Rate”, for more details see section 4.5.

⁴ the natches on the curves of the figure 3 correspond to 4 times the variance of the results ($\pm 2\sigma$).

⁵ The test set include 1613 examples and the training set 3783 examples.

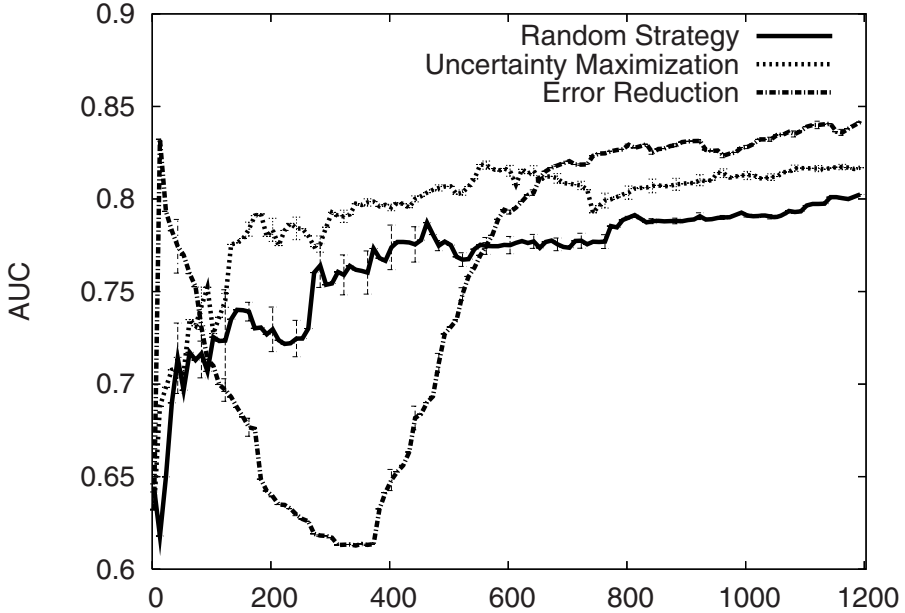


Fig. 3. Focus of the results on the test using [0:1200] training examples

a single class. Consequently, we use the sum of the AUCs weighted by reference class’s prevalence in the data.

The “risk reduction” is the strategy which maximizes the quality of the model for a number of training examples in the range [2:100]. Between 100 and 700 the strategy based on uncertainty wins. After 600 training examples the three strategies converge to the optimal AUC (Area Under Roc Curve).

The two active strategies allow obtaining faster than the random strategy the optimal result (the optimal AUC is 0.84 using the whole training set). The use of active learning is positive in this real problem. However the results obtained raise questions which will be detailed in the next section.

5 Discussion and Conclusion

This paper shows the interest of active learning for a field where acquisition and labeling of data are particularly expensive. Obtained results show that active learning is relevant for the detection of expressed emotions in speech. But whatever the strategy considered (even the two strategies evoked in section 3.4 but not detailed in this paper) several questions exist and can be raised:

- **evaluation** - The quality of an active strategy is usually represented by a curve assessing the performance of the model versus the number of training examples labeled (see Figure 3). The performance criterion used can take

several different ways according to the problem. This type of curve allows only comparisons between strategies in a punctual way, i.e. for a point on the curve (a given number of training examples). If two curves pass each other (as in the Figure 3, it is impossible to determine if a strategy is better than another (on the total set of training examples). The elaboration of a criterion which measures the contribution of a strategy compared to the random strategy on the whole data set should be interesting. This point will be discussed in a future paper.

- **test set** - Active learning strategies are, often, used when data acquisition is expensive. Therefore, in practice, a test set is not available (otherwise it can be used to the training) and the evaluation of the model during a strategy is difficult to implement.
- **stopping criterion** - The maximal number of examples to labeled, or an estimation of the progress of the model, can be used to stop the algorithm. This is very link to the use of a test set or the model employed. For example in the Figure 3 the strategy based on the risk gives the same results when 15 examples have been labeled than results using all the available data. In this case the cost using 15 examples and 600 will be not the same... A good criterion should be independent of the model and of a test set. Actual experiments (not yet published) will allow us to propose a criterion of this type at the end of 2007.
- **number of examples to be labeled** - the state of the art seems to incorporate an only one example at each step of the strategy. But in real case the expert is a human and when the model needs time to learn at each iteration of the strategy this could be not efficient. Sometimes more than one example must be incorporated. This aspect has been a little bit studied in [26] but it has to be more analyse in the future.
- **uncertain environment** - If an answer could be given to the points above then active strategies could be used for on-line learning in an uncertain environment. For example to tag part of graph (graph here is social network). When writing this paper we hope that this point will accepted and incorporated in a proposition sent to a French Project (and financed by the French National Agency of Research (ANR)) grouping industrial and universities.

Generally, active learning strategies estimate the utility of training examples. These criteria could be used for on-line training. The training set would be consisted of the N examples the more “useful” seen until now (with N fixed). This approach would be able to consider non stationary problems and it is able to train a model which adapts itself to the variations of the observed system.

For the detection of expressed emotions in speech could be treated by a double strategy reducing the cost linked to the data : (i) a variables selection allowing to preserve only the necessary and sufficient characteristics for classification; (ii) an examples selection allowing to preserve only useful instances for training. This will be explored in future work.

References

1. Harmon, M.: Reinforcement learning: a tutorial (1996)
<http://eureka1.aa.wpafb.af.mil/rltutorial/>
2. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. *ACM Computing Surveys* 31(3), 264–323 (1999)
3. Jamy, I., Jen, T.-Y., Laurent, D., Loizou, G., Sy, O.: Extraction de règles d’association pour la prédiction de valeurs manquantes. *Revue Africaine de la Recherche en Informatique et Mathématique Appliquée ARIMA Spécial CARI04*, 103–124 (2005)
4. Chapelle, O., Schölkopf, B., Zien, A.: *Semi-Supervised Learning*. MIT Press, Cambridge, MA (in press) (2006). http://www.kyb.tuebingen.mpg.de/ssl-book/ssl_toc.pdf
5. Cohn, D., Caruana, R., McCallum, A.: Semi-supervised clustering with user feedback. Technical Report 1892, Cornell University (2003)
6. Chapelle, O., Zien, A.: Semi-supervised classification by low density separation. In: *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics* (2005)
7. Castro, R., Willett, R., Nowak, R.: Faster rate in regression via active learning. In: *NIPS (Neural Information Processing Systems)*, Vancouver (2005)
8. Singh, A., Nowak, R., Ramanathan, P.: Active learning for adaptive mobile sensing networks. In: *IPSN ’06. Proceedings of the fifth international conference on Information processing in sensor networks*, pp. 60–68. ACM Press, New York (2006)
9. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: *Proc. 18th International Conf. on Machine Learning*, pp. 441–448. Morgan Kaufmann, San Francisco (2001)
10. Muslea, I.: *Active Learning With Multiple View*. Phd thesis, University of southern california (2002)
11. Lewis, D., Gale, A.: A sequential algorithm for training text classifiers. In: Croft, W.B., van Rijsbergen, C.J. (eds.) *Proceedings of SIGIR-94. 17th ACM International Conference on Research and Development in Information Retrieval*, Dublin. LNCS, pp. 3–12. Springer, Heidelberg (1994)
12. Thrun, S.B., Möller, K.: Active exploration in dynamic environments. In: Moody, J.E., Hanson, S.J., Lippmann, R.P. (eds.) *Advances in Neural Information Processing Systems*, vol. 4, pp. 531–538. Morgan Kaufmann Publishers, San Francisco (1992)
13. Cohn, D.A., Ghahramani, Z., Jordan, M.I.: Active learning with statistical models. In: Tesauro, G., Touretzky, D., Leen, T. (eds.) *Advances in Neural Information Processing Systems*, vol. 7, pp. 705–712. The MIT Press, Cambridge (1995)
14. Zhu, X., Lafferty, J., Ghahramani, Z.: Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In: *ICML (International Conference on Machine Learning)*, Washington (2003)
15. Parzen, E.: On estimation of a probability density function and mode. *Annals of Mathematical Statistics* 33, 1065–1076 (1962)
16. Freund, Y., Seung, H.S., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. *Machine Learning* 28(2-3), 133–168 (1997)
17. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: *Computational Learning Theory*, pp. 287–294 (1992)
18. Dasgupta, S.: Analysis of greedy active learning strategy. In: *NIPS (Neural Information Processing Systems)*, San Diego (2005)

19. Cohn, D.A., Atlas, L., Ladner, R.E.: Improving generalization with active learning. *Machine Learning* 15(2), 201–221 (1994)
20. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. In: Langley, P. (ed.) *Proceedings of ICML-00. 17th International Conference on Machine Learning*, Stanford, US, pp. 999–1006. Morgan Kaufmann Publishers, San Francisco (2000)
21. Liscombe, J., Riccardi, G., Hakkani-Tür, D.: Using context to improve emotion detection in spoken dialog systems. In: *InterSpeech*, Lisbon (2005)
22. Poulain, B.: Sélection de variables et modélisation d'expressions d'émotions dans des dialogues hommes-machine (in french). In: *EGC (Extraction et Gestion de Connaissance)* (2006), Lille. + Technical Report available here: <http://perso.rd.francetelecom.fr/lemaire>
23. Boullé, M.: An enhanced selective naïve bayes method with optimal discretization. In: Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L. (eds.) *Feature extraction, foundations and Application*, August 2006. LNCS, pp. 499–507. Springer, Heidelberg (2006)
24. Chappelle, O.: Active learning for parzen windows classifier. In: *AI & Statistics*, Barbados, pp. 49–56 (2005)
25. Guide, V., Rakotomamonjy, C.S.: Méthode à noyaux pour l'identification d'émotion. In: *RFIA (Reconnaissance des Formes et Intelligence Artificielle)* (2003)
26. Bondu, A., Lemaire, V.: Etude de l'influence du nombre d'exemples à étiqueter dans une procédure d'apprentissage actif. In: *CAP 2006 (Conference francophone sur l'apprentissage automatique)* (submitted to, 2006)

Neural Business Control System

M. Lourdes Borrajo¹, Juan M. Corchado², E.S. Corchado³, and M.A. Pellicer³

¹ Dept. Informática, University of Vigo,

Esc. Superior de Ingeniería Informática, Edificio Politécnico,
Campus Universitario As Lagoas s/n, 32004, Ourense, Spain

² Departamento de Informática y Automática, University of Salamanca,
Plaza de la Merced s/n, 37008 Salamanca, Spain

³ Dept. de Ingeniería Civil, University of Burgos,
Esc. Politécnica Superior, Edificio C, C/ Francisco de Vitoria, Burgos, Spain

Abstract. The firms have need of a control mechanism in order to analyse whether they are achieving their goals. A tool that automates the business control process has been developed based on a case-based reasoning system. The objective of the system is to facilitate the process of internal auditing. The system analyses the data that characterises each one of the activities carried out by the firm, then determines the state of each activity and calculates the associated risk. This system uses a different problem solving method in each of the steps of the reasoning cycle. A Maximum Likelihood Hebbian Learning-based method that automates the organization of cases and the retrieval stage of case-based reasoning systems is presented in this paper. The proposed methodology has been derived as an extension of the Principal Component Analysis, and groups similar cases, identifying clusters automatically in a data set in an unsupervised mode. The system has been tested in 10 small and medium companies in the textile sector, located in the northwest of Spain and the results obtained have been very encouraging.

1 Introduction

The firms need a control mechanism in order to analyse whether they are achieving their goals, being based on a series of organizational policies and specific procedures. This group of policies and procedures are named "controls", and they all conform to the structure of business control of the company. Therefore, periodic internal audits are necessary. Nevertheless the firms are characterized by their great dynamism. It is necessary to construct models that facilitate the analysis of work carried out in changing environments, such as finance.

The processes carried out inside a firm are grouped in functional areas [5] denominated "Functions". A Function is a group of coordinated and related activities, which are necessary to reach the objectives of the firm and are carried out in a systematic and iterative way [22]. The functions that are usually carried out within a firm are: Purchases, Cash Management, Sales, Information Technology, Fixed Assets Management, Compliance to Legal Norms and Human Resources. In turn, each one of these functions is broken down into a series of activities. Each activity is composed

of a number of tasks. Control procedures have also to be established in the tasks to ensure that the established objectives are achieved.

The objective of the developed system is to identify the state or situation of each one of activities of the company and to calculate the risk associated with this state. The system is implemented using a case-based reasoning (CBR) system [1, 19, 27, 21]. The CBR system uses different problem solving techniques [14, 23].

This paper presents a Maximum Likelihood Hebbian Learning (MLHL) based model to automate the process of case indexing and retrieval, which may be used in problems in which the cases are characterised predominantly by numerical information.

Maximum Likelihood Hebbian Learning (MLHL) based models were first developed as an extension of Principal Component Analysis [24, 25]. Maximum Likelihood Hebbian Learning Based Method attempts to identify a small number of data points, which are necessary to solve a particular problem to the required accuracy. These methods have been successfully used in the unsupervised investigation of structure in data sets [3, 4]. We have previously investigated the use of Artificial Neural Networks [8] and Kernel Principal Component Analysis (KPCA) [11, 13] to identify cases, which will be used in a case based reasoning system. In this paper, we present a novel hybrid technique. The ability of the Maximum Likelihood Hebbian Learning-based methods presented in this paper to cluster cases/instances and to associate cases to clusters can be used to successfully prune the case-base without losing valuable information.

This paper first presents the Maximum Likelihood Hebbian Learning Based Method and its theoretical background. We review Principal Component Analysis (PCA) which has been the most frequently reported linear operation involving unsupervised learning for data compression, which aims to find that orthogonal basis which maximises the data's variance for a given dimensionality of basis. Then, the Exploratory Projection Pursuit (EPP) theory is outlined. It is shown how Maximum Likelihood Hebbian Learning Based Method may be derived from PCA and it could be viewed as a method of performing EPP. Then, the proposed CBR based model is presented. The system results are evaluated and, finally, the conclusions are presented.

2 Maximum Likelihood Hebbian Learning Based Method

The use of Maximum Likelihood Hebbian Learning Based Method has been derived from the work of [4, 11, 12, 13], etc. in the field of pattern recognition as an extension of Principal Component Analysis (PCA) [24, 25].

2.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a standard statistical technique for compressing data; it can be shown to give the best linear compression of the data in terms of least mean square error. There are several artificial neural networks which have been shown to perform PCA e.g. [24, 25]. We will apply a negative feedback implementation [10].

The basic PCA network is described by equations (1)-(3). Let us have an N-dimensional input vector at time t , $\mathbf{x}(t)$, and an M-dimensional output vector, \mathbf{y} , with W_{ij} being the weight linking input j to output i . η is a learning rate. Then the activation passing and learning is described by

$$\text{Feedforward: } y_i = \sum_{j=1}^N W_{ij} x_j, \forall i \tag{1}$$

$$\text{Feedback: } e_j = x_j - \sum_{i=1}^M W_{ij} y_i \tag{2}$$

$$\text{Change weights: } \Delta W_{ij} = \eta e_j y_i \tag{3}$$

We can readily show that this algorithm is equivalent to Oja’s Subspace Algorithm [24]:

$$\Delta W_{ij} = \eta e_j y_i = \eta (x_j - \sum_k W_{kj} y_k) y_i \tag{4}$$

and so this network not only causes convergence of the weights but causes the weights to converge to span the subspace of the Principal Components of the input data. We might ask then why we should be interested in the negative feedback formulation rather than the formulation (4) in which the weight change directly uses negative feedback. The answer is that the explicit formation of residuals (2) allows us to consider probability density functions of the residuals in a way which would not be brought to mind if we use (4).

Exploratory Projection Pursuit (EPP) is a more recent statistical method aimed at solving the difficult problem of identifying structure in high dimensional data. It does this by projecting the data onto a low dimensional subspace in which we search for its structure by eye. However not all projections will reveal the data's structure equally well. We therefore define an index that measures how “interesting” a given projection is, and then represent the data in terms of projections that maximise that index.

The first step in our exploratory projection pursuit is to define which indices represent interesting directions. Now “interesting” structure is usually defined with respect to the fact that most projections of high-dimensional data onto arbitrary lines through most multi-dimensional data give almost Gaussian distributions [7]. Therefore if we wish to identify “interesting” features in data, we should look for those directions onto which the data-projections are as far from the Gaussian as possible.

It was shown in [17] that the use of a (non-linear) function creates an algorithm to find those values of W which maximise that function whose derivative is $f()$ under the constraint that W is an orthonormal matrix. This was applied in [10] to the above network in the context of the network performing an Exploratory Projection Pursuit. Thus if we wish to find a direction which maximises the kurtosis of the distribution which is measured by s_4 , we will use a function $f(s) \approx s_3$ in the algorithm. If we wish to find that direction with maximum skewness, we use a function $f(s) \approx s_2$ in the algorithm.

2.2 ϵ -Insensitive Hebbian Learning

It has been shown [29] that the nonlinear PCA rule

$$\Delta W_{ij} = \eta \left(x_j f(y_i) - f(y_i) \sum_k W_{kj} f(y_k) \right) \tag{5}$$

can be derived as an approximation to the best non-linear compression of the data. Thus we may start with a cost function

$$J(W) = 1^T E \left\{ (\mathbf{x} - Wf(W^T \mathbf{x}))^2 \right\} \tag{6}$$

which we minimise to get the rule (5). [20] used the residual in the linear version of (6) to define a cost function of the residual

$$J = f_1(\mathbf{e}) = f_1(\mathbf{x} - W\mathbf{y}) \tag{7}$$

where $f_1 = \|\cdot\|^2$ is the (squared) Euclidean norm in the standard linear or nonlinear PCA rule. With this choice of $f_1(\cdot)$, the cost function is minimised with respect to any set of samples from the data set on the assumption that the residuals are chosen independently and identically distributed from a standard Gaussian distribution. We may show that the minimisation of J is equivalent to minimising the negative log probability of the residual, \mathbf{e} , if \mathbf{e} is Gaussian.

$$\text{Let } p(\mathbf{e}) = \frac{1}{Z} \exp(-\mathbf{e}^2) \tag{8}$$

Then we can denote a general cost function associated with this network as

$$J = -\log p(\mathbf{e}) = (\mathbf{e})^2 + K \tag{9}$$

where K is a constant. Therefore performing gradient descent on J we have

$$\Delta W \propto -\frac{\partial J}{\partial W} = -\frac{\partial J}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\partial W} \approx \mathbf{y}(2\mathbf{e})^T \tag{10}$$

where we have discarded a less important term. See [17] for details.

In general [26], the minimisation of such a cost function may be thought to make the probability of the residuals greater dependent on the probability density function (pdf) of the residuals. Thus if the probability density function of the residuals is known, this knowledge could be used to determine the optimal cost function. [13] investigated this with the (one dimensional) function:

$$p(\mathbf{e}) = \frac{1}{2 + \epsilon} \exp(-|\mathbf{e}|_\epsilon) \tag{11}$$

$$\text{where } |e|_\varepsilon = \begin{cases} 0 & \forall |e| < \varepsilon \\ |e| - \varepsilon & \text{otherwise} \end{cases} \tag{12}$$

with ε being a small scalar ≥ 0 .

Fyfe and MacDonald [13] described this in terms of noise in the data set. However, we feel that it is more appropriate to state that, with this model of the pdf of the residual, the optimal $f_1(\cdot)$ function is the ε -insensitive cost function:

$$f_1(\mathbf{e}) = |\mathbf{e}|_\varepsilon \tag{13}$$

In the case of the negative feedback network, the learning rule is

$$\Delta W \propto -\frac{\partial J}{\partial W} = -\frac{\partial f_1(\mathbf{e})}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\partial W} \tag{14}$$

which gives:

$$\Delta W_{ij} = \begin{cases} 0 & \text{if } |e_j| < \varepsilon \\ \eta y(\text{sign}(e)) & \text{otherwise} \end{cases} \tag{15}$$

The difference with the common Hebb learning rule is that the sign of the residual is used instead the value of the residual. Because this learning rule is insensitive to the magnitude of the input vectors \mathbf{x} , the rule is less sensitive to outliers than the usual rule based on mean squared error. This change from viewing the difference after feedback as simply a residual rather than an error permits us to consider a family of cost functions each member of which is optimal for a particular probability density function associated with the residual.

2.3 Applying Maximum Likelihood Hebbian Learning

The Maximum Likelihood Hebbian Learning algorithm is constructed now on the bases of the previously presented concepts as outlined here. Now the ε -insensitive learning rule is clearly only one of a possible family of learning rules which are suggested by the family of exponential distributions. This family was called an exponential family in [16] though statisticians use this term for a somewhat different family. Let the residual after feedback have probability density function

$$p(\mathbf{e}) = \frac{1}{Z} \exp(-|\mathbf{e}|^p) \tag{16}$$

Then we can denote a general cost function associated with this network as

$$J = E(-\log p(\mathbf{e})) = E(|\mathbf{e}|^p + K) \tag{17}$$

where K is a constant independent of W and the expectation is taken over the input data set. Therefore performing gradient descent on J we have

$$\Delta W \propto -\frac{\partial J}{\partial W} \Big|_{W(t-1)} = -\frac{\partial J}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\partial W} \Big|_{W(t-1)} \approx E\{\mathbf{y}(p|\mathbf{e}|^{p-1} \text{sign}(\mathbf{e}))^T \Big|_{W(t-1)}\} \quad (18)$$

where T denotes the transpose of a vector and the operation of taking powers of the norm of \mathbf{e} is on an element wise basis as it is derived from a derivative of a scalar with respect to a vector.

Computing the mean of a function of a data set (or even the sample averages) can be tedious, and we also wish to cater for the situation in which samples keep arriving as we investigate the data set and so we derive an online learning algorithm. If the conditions of stochastic approximation [18] are satisfied, we may approximate this with a difference equation. The function to be approximated is clearly sufficiently smooth and the learning rate can be made to satisfy $\eta_k \geq 0, \sum_k \eta_k = \infty, \sum_k \eta_k^2 < \infty$ and so we have the rule:

$$\Delta W_{ij} = \eta \cdot y_i \cdot \text{sign}(e_j) |e_j|^{p-1} \quad (19)$$

We would expect that for leptokurtotic residuals (more kurtotic than a Gaussian distribution), values of $p < 2$ would be appropriate, while for platykurtotic residuals (less kurtotic than a Gaussian), values of $p > 2$ would be appropriate. Researchers from the community investigating Independent Component Analysis [15, 16] have shown that it is less important to get exactly the correct distribution when searching for a specific source than it is to get an approximately correct distribution i.e. all supergaussian signals can be retrieved using a generic leptokurtotic distribution and all subgaussian signals can be retrieved using a generic platykurtotic distribution. Our experiments will tend to support this to some extent but we often find accuracy and speed of convergence are improved when we are accurate in our choice of p . Therefore the network operation is:

$$\text{Feedforward: } y_i = \sum_{j=1}^N W_{ij} x_j, \forall_i \quad (20)$$

$$\text{Feedback: } e_j = x_j - \sum_{i=1}^M W_{ij} y_i \quad (21)$$

$$\text{Weights change: } \Delta W_{ij} = \eta \cdot y_i \cdot \text{sign}(e_j) |e_j|^{p-1} \quad (22)$$

Fyfe and MacDonald [13] described their rule as performing a type of PCA, but this is not strictly true since only the original (Oja) ordinary Hebbian rule actually performs PCA. It might be more appropriate to link this family of learning rules to Principal Factor Analysis since PFA makes an assumption about the noise in a data set and then removes the assumed noise from the covariance structure of the data before performing a PCA. We are doing something similar here in that we are basing our PCA-type rule on the assumed distribution of the residual. By maximising the likelihood of the residual with respect to the actual distribution, we are matching the learning rule to the probability density function of the residual.

More importantly, we may also link the method to the standard statistical method of Exploratory Projection Pursuit: now the nature and quantification of the interestingness is in terms of how likely the residuals are under a particular model of the probability density function of the residuals. In the results reported later, we also sphere the data before applying the learning method to the sphered data and show that with this method we may also find interesting structure in the data.

2.4 Sphering of the Data

Because a Gaussian distribution with mean a and variance x is no more or less interesting than a Gaussian distribution with mean b and variance y - indeed this second order structure can obscure higher order and more interesting structure - we remove such information from the data. This is known as "sphering". That is, the raw data is translated till its mean is zero, projected onto the principal component directions and multiplied by the inverse of the square root of its eigenvalue to give data which has mean zero and is of unit variance in all directions. So for input data X we find the covariance matrix.

$$\Sigma = \langle (X - \langle X \rangle)(X - \langle X \rangle)^T \rangle = UDU^T \quad (23)$$

Where U is the eigenvector matrix, D the diagonal matrix of eigenvalues, T denotes the transpose of the matrix and the angled brackets indicate the ensemble average. New samples, drawn from the distribution are transformed to the principal component axes to give y where

$$y_i = \frac{1}{\sqrt{D_i}} \sum_{j=1}^n U_{ij} (X_i - \langle X_i \rangle), \text{ for } 1 \leq i \leq m \quad (24)$$

Where n is the dimensionality of the input space and m is the dimensionality of the sphered data.

3 Phases of the Proposed System

This section describes the business control system in detail. Although the aim is to develop a generic model useful in any type of small to medium enterprise, the initial work has focused in the textile sector to facilitate the research and its evaluation. The model here presented may be extended or adapted for other sectors. Twenty two companies from the North-west of Spain have collaborated in this research, working mainly for the Spanish market. The companies have different levels of automation and all of them were very interested in a tool such as the one developed in the framework of this investigation. After analyzing the data relative to the activities developed within a given firm, the constructed system is able to determine the state of each of the activities and calculate the associated risk. The problem solving mechanism developed takes its decision using the help of a CBR system whose memory has been fed with cases constructed with information provided by the firm

and with prototypical cases identified by 34 business control experts who have collaborated and supervised the model developed.

The cycle of operations of the developed case based reasoning system is based on the classic life cycle of a CBR system [1, 28]. A case represents the “shape” of a given activity developed in the company. Each case is composed of the following attributes:

- *Case number*: Unique identification: positive integer number.
- *Input vector*: Information about the tasks (n sub-vectors) that constitute an industrial activity: $((IR_1, V_1), (IR_2, V_2), \dots, (IR_n, V_n))$ for n tasks. Each task sub-vector has the following structure (IR_i, V_i) :
 - IR_i : importance rate for this task within the activity. It can only take one of the following values: VHI (Very high importance), HI (High Importance), AI (Average Importance), LI (Low Importance), VLI (Very low importance)
 - V_i : Value of the realization state of a given task: a positive integer number (between 1 and 10).
- *Function number*: Unique identification number for each function
- *Activity number*: Unique identification number for each activity
- *Reliability*: Percentage of probability of success. It represents the percentage of success obtained using the case as a reference to generate recommendations.
- *Degree of membership*: $((n_1, \mu_1), (n_2, \mu_2), \dots, (n_k, \mu_k))$
 - n_i : represents the i^{th} cluster
 - μ_i : represents the membership value of the case to the cluster n_i
- *Activity State*: degree of perfection of the development of the activity, expressed by percentage.

Every time that it is necessary to obtain a new estimate of the state of an activity, the system evolves through several phases. This evolution allows the system, on the one hand, (i) to identify the latest situations most similar to the current situation, (ii) to adapt the current knowledge to generate an estimate of the risk of the activity being analysed. The following sections describe the different phases of the proposed model.

3.1 Evaluation of the State of the Activity

For each activity to analyse, the system uses the data for this activity, introduced by the firm’s internal auditor, to construct the problem case. For each task making up the activity analyzed, the problem case is composed of the value of the realization state for that task, and its level of importance within the activity (according to the internal auditor).

In the retrieval step, the system retrieves K cases – the most similar cases to the problem case. This is done with the Maximum Likelihood Hebbian Learning proposed method. Applying equations 20 to 22 to the case-base, the MLHL algorithm groups the cases in clusters automatically. The proposed indexing mechanism classifies the cases/instances automatically, clustering together those of similar structure. One of the great advantages of this technique is that it is an unsupervised method so we do not need to have any information about of the data before hand. When a new case is presented to the CBR system, it is identified as belonging to a

particular type by applying also equations 20 to 22 to it. This mechanism may be used as a universal retrieval and indexing mechanism to be applied to any problem similar to the presented here.

Maximum Likelihood Hebbian Learning techniques are used because of the size of the database and the need to group the most similar cases together in order to help retrieve the cases that most resemble the given problem.

Maximum Likelihood Hebbian Learning techniques are especially interesting for non-linear or ill-defined problems, making it possible to treat tasks involved in the processing of massive quantities of redundant or imprecise information. It allows the available data to be grouped into clusters with fuzzy boundaries, expressing uncertain knowledge.

The following step, the re-use phase, aims to obtain an initial estimation of the state of the activity analysed using a RBF networks are used [9, 6, 8]. As in the previous stage, the number of attributes of the problem case depends on the activity analyzed. Therefore it is necessary to establish an RBF network system, one for each of the activities to be analysed.

The retrieved K cases are used by the RBF network as a training group that allows it to adapt its configuration to the new problem encountered before generating the initial estimation.

The RBF network is characterized by its ability to adapt, to learn rapidly, and to generalize (especially in interpolation tasks). Specifically, within this system the network acts as a mechanism capable of absorbing knowledge about a certain number of cases and generalizing from them. During this process, the RBF network, interpolates and carries out predictions without forgetting part of those already carried out. The system's memory acts as a permanent memory capable of maintaining many cases or experiences while the RBF network acts as a short term memory, able to recognize recently learnt patterns and to generalize from them.

The objective of the revision phase is to confirm or refute the initial solution proposed by the RBF network, thereby obtaining a final solution and calculating the control risk. In view of the initial estimation or solution generated by the RBF network, the internal auditor will be responsible for deciding if the solution is accepted. For this it is based on the knowledge he/she retains, specifically, knowledge about the company with which he/she is working. If he/she considers that the estimation given is valid, the system will take the solution as the final solution and in the following phase of the CBR cycle, a new case will be stored in the case base consisting of the problem case and the final solution. The system will assign the case an initial reliability of 100%. If on the other hand, the internal auditor considers the solution given by the system to be invalid, he will give his own solution which the system will take as the final solution and which together with the problem case will form the new case to be stored in the case base in the following phase. This new case will be given a reliability of 30%. This value has been defined taking into account the opinion of various auditors in terms of the weighting that should be assigned to the personal opinion of the internal auditor.

From the final solution: state of activity, the system calculates the control risk associated with the activity. Every activity developed in the business sector has a risk associated with it that indicates the negative influence that affects the good operation of the firm. In other words, the control risk of an activity measures the impact that the

current state of the activity has on the business process as a whole. In this study, the level of risk is valued at three levels: low, medium and high. The calculation of the level of control risk associated with an activity is based on the current state of the activity and its level of importance. This latter value was obtained after analysing data obtained from a series of questionnaires (98 in total) carried out by auditors throughout Spain. In these questionnaires the auditors were asked to rate subjects from 1-10 according to the importance or weighting of each activity in terms of the function that it belonged to. The higher the importance of the activity, the greater its weighting within the business control system. The level of control risk was then calculated from the level of importance given to the activity by the auditors and the final solution obtained after the revision phase. For this purpose, if-then rules are employed.

The last phase of the system is the incorporation of the system's memory of what has been learnt after resolving a new problem. Once the revision phase has been completed, after obtaining the final solution, a new case (*problem + solution*) is constructed, which is stored in the system's memory. Apart from the overall knowledge update involving the insertion of a new case within the system memory, the hybrid system presented carries out a local adaptation of the knowledge structures that it uses.

Maximum Likelihood Hebbian Learning technique contained within the prototypes related to the activity corresponding to the new case is reorganised in order to respond to the appearance of this new case, modifying its internal structure and adapting itself to the new knowledge available.

The RBF network uses the new case to carry out a complete learning cycle, updating the position of its centres and modifying the value of the weightings that connect the hidden layer with the output layer.

4 Results and Conclusions

A complete set of tests has been carried out over a total amount of 10 small to medium companies. From the total number of 10, 6 were medium-sized, while 4 were small sized firms, all of them pertaining to the textile sector. Spanish auditors performed 98 surveys in order to obtain the data that would feed the process of generating the prototype cases, needed to build the system's base classes. Another 34 surveys were carried out by different experts of functional areas of firms within this sector.

For a given company, each one of its activities was evaluated by the system, obtaining a level of risk. On the other hand, we request to five external and independent auditors that they analyzed the situation of each company. The mission of the auditors is to estimate the state of each activity, the same as the proposed system makes. Then, we compare the result of the evaluation obtained by the auditors with the result obtained by the system. Figure 1 shows the differences between the results obtained by the system and the external auditors about the function "Information Technology". It can be observed that the results obtained by the system are very similar to those obtained by the external auditors.

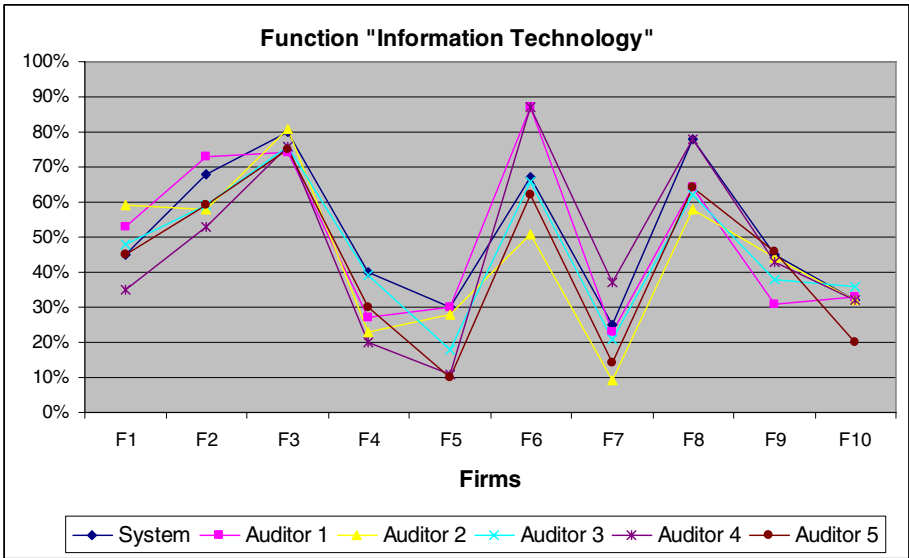


Fig. 1. Obtained results

In general, it could be said that these results demonstrate the suitability of the techniques used for their integration in the developed intelligent control system.

This article presents a neuro-symbolic system that use a CBR system employed as a basis for hybridization of a Maximum Likelihood Hebbian Learning technique, and a RBF net.

The used reasoning model can be applied in situations that satisfy the following conditions:

1. Each problem can be represented in the form of a vector of quantified values.
2. The case base should be representative of the total spectrum of the problem.
3. Cases must be updated periodically.
4. Enough cases should exist to train the net.

The prototype cases used for the construction of the case base are artificial and have been created from surveys carried out with auditors and experts in different functional areas. The system is able to estimate or identify the state of the activities of the firm and their associated risk.

Estimation in the environment of firms is difficult due to the complexity and the great dynamism of this environment. However, the developed model is able to estimate the state of the firm with precision. The system will produce better results if it is fed with cases related to the sector in which it will be used. This is due to the dependence that exists between the processes in the firms and the sector where the company is located. Future experiments will help to identify how the constructed prototype will perform in other sectors and how it will have to be modified in order to improve its performance. We have demonstrated a new technique for case indexing and retrieval, which could be used to construct case-based reasoning systems. The

basis of the method is a Maximum Likelihood Hebbian Learning algorithm. This method provides us with a very robust model for indexing the data and retrieving instances without any need of information about the structure of the data set.

References

1. Aamodt, A., Plaza, E.: Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches. *AICOM* 7(1) (1994)
2. Borrajo, L.: Sistema híbrido inteligente aplicado a la auditoría de los sistemas internos. Phd Thesis. Teses de doutoramento da Universidade de Vigo. Universidade de Vigo (Spain). ISBN: 84-8158-274-3 (December 2003)
3. Corchado, E., Fyfe, C.: Maximum and Minimum Likelihood Hebbian Rules as a Exploratory Method. 9th International Conference on Neural Information Processing, November 18-22, 2002, Singapore (2002)
4. Corchado, E., MacDonald, D., Fyfe, C.: Optimal Projections of High Dimensional Data. In: *ICDM '02. The 2002 IEEE International Conference on Data Mining*, Maebashi TERRSA, Maebashi City, December 9-12, 2002, IEEE Computer Society, Los Alamitos (2002)
5. Corchado, J.M., Borrajo, L., Pellicer, M.A., Yáñez, J.C.: Neuro-symbolic System for Business Internal Control. In: Perner, P. (ed.) *ICDM 2004. LNCS (LNAI)*, vol. 3275, pp. 302–9743. Springer, Heidelberg (2004)
6. Corchado, J.M., Díaz, F., Borrajo, L., Fdez-Riverola, F.: Redes Neuronales Artificiales: Un enfoque práctico. Departamento de publicaciones de la Universidad de Vigo (2000)
7. Diaconis, P., Freedman, D.: Asymptotics of Graphical Projections. *The Annals of Statistics* 12(3), 793–815 (1984)
8. Fdez-Riverola, F., Corchado, J.M.: FSfRT: Forecasting System for Red Tides. *Applied Intelligence. Special Issue on Soft Computing in Case-Based Reasoning* 21(3), 251–264 (2004) ISSN 0924-669X
9. Fritzke, B.: Fast Learning with Incremental RBF Networks. *Neural Processing Letters* 1(1), 2–5 (1994)
10. Fyfe, C., Baddeley, R.: Non-linear data structure extraction using simple Hebbian networks. *Biological Cybernetics* 72(6), 533–541 (1995)
11. Fyfe, C., Corchado, E.: Maximum Likelihood Hebbian Rules. 10th European Symposium on Artificial Neural Networks, ESANN'2002, Bruges, April 24-25-26 (2002a)
12. Fyfe, C., Corchado, E.: A New Neural Implementation of Exploratory Projection Pursuit. In: Yin, H., Allinson, N.M., Freeman, R., Keane, J.A., Hubbard, S. (eds.) *IDEAL 2002. LNCS*, vol. 2412, pp. 12–14. Springer, Heidelberg (2002b)
13. Fyfe, C., MacDonald, D.: ϵ -Insensitive Hebbian learning, *Neuro Computing* (2001)
14. Hunt, J., Miles, R.: Hybrid case-based reasoning. *The Knowledge Engineering Review* 9(4), 383–397 (1994)
15. Hyvärinen, A.: Complexity Pursuit: Separating interesting components from time series. *Neural Computation* 13, 883–898 (2001)
16. Hyvärinen, A., Karhunen, J., Oja, E.: *Independent Component Analysis*. Wiley, Chichester (2002)
17. Karhunen, J., Joutsensalo, J.: Representation and Separation of Signals Using Non-linear PCA Type Learning. *Neural Networks* 7, 113–127 (1994)

18. Kashyap, R.L., Blaydon, C.C., Fu, K.S.: Stochastic Approximation. In: Mendel, J.M. (ed.) *A Prelude to Neural Networks: Adaptive and Learning Systems*, Prentice Hall, Englewood Cliffs (1994) ISBN 0-13-147448-0
19. Kolodner, J.: *Case-Based Reasoning*. Morgan Kaufmann, San Francisco (1993)
20. Lai, P.L., Charles, D., Fyfe, C.: Seeking Independence using Biologically Inspired Artificial Neural Networks. In: Girolami, M.A. (ed.) *Developments in Artificial Neural Network Theory: Independent Component Analysis and Blind Source Separation*, Springer, Heidelberg (2000)
21. Lenz, M., Bartsch-Spörl, B., Burkhard, H.-D., Wess, S. (eds.): *Case-Based Reasoning Technology*. LNCS (LNAI), vol. 1400. Springer, Heidelberg (1998)
22. Mas, J., Ramió, C.: *La Auditoría Operativa en la Práctica*. Ed. Marcombo, Barcelona (1997)
23. Medsker, L.R.: *Hybrid Intelligent Systems*. Kluwer Academic Publishers, Dordrecht (1995)
24. Oja, E.: Neural Networks, Principal Components and Subspaces. *International Journal of Neural Systems* 1, 61–68 (1989)
25. Oja, E., Ogawa, H., Wangviwattana, J.: Principal Components Analysis by Homogeneous Neural Networks, part 1, The Weighted Subspace Criterion. *IEICE Transaction on Information and Systems* E75D, 366–375 (1992)
26. Smola, A.J., Scholkopf, B.: A Tutorial on Support Vector Regression. Technical Report NC2-TR-1998-030, NeuroCOLT2 Technical Report Series (1998)
27. Watson, I.: *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann, San Francisco (1997)
28. Watson, I., Marir, F.: Case-Based Reasoning: A Review. *The Knowledge Engineering Review* 9(4), 355–381 (1994)
29. Xu, L.: Least Mean Square Error Reconstruction for Self-Organizing Nets. *Neural Networks* 6, 627–648 (1993)

A Framework for Discovering and Analyzing Changing Customer Segments

Mirko Böttcher, Martin Spott, and Detlef Nauck

Intelligent Systems Research Centre, BT Group plc
Adastral Park, IP5 3RE Ipswich, United Kingdom
{Mirko.Boettcher, Martin.Spott, Detlef.Nauck}@bt.com

Abstract. Identifying customer segments and tracking their change over time is an important application for enterprises who need to understand what their customers expect from them. Customer segmentation is typically done by applying some form of cluster analysis. In this paper we present an alternative approach based on association rule mining and a notion of interestingness. Our approach allows us to detect arbitrary segments and analyse their temporal development. Our approach is assumption-free and pro-active and can be run continuously. Newly discovered segments or relevant changes will be reported automatically based on the application of an interestingness measure.

1 Introduction

Businesses, especially in the service industry, need to understand their customers in order to serve them best. Understanding customers involves collecting as much data as possible about interactions between customers and the business, analyse this data to turn it into information and finally learn from it and take action. This process is supported by techniques from data warehousing, data quality management, knowledge discovery in databases (or data mining), business intelligence, business process management etc. In this paper we will look at a particular aspect of the analytical process – the discovery of changing customer segments.

When we hear about customer segments we typically think about marketing-driven demographic groups that are defined using a great deal of domain understanding. This approach requires typically running extensive surveys on a significant part of the customer base to learn about their preferences, views, standard of living, consumer behavior etc. Based on domain understanding a number of segments are then identified and customers are assigned to segments based on some similarity measure. Typically, approaches from cluster analysis are used to initially identify groups in the data which are then interpreted as potential customer segments. The whole process is based on manual analysis and is typically expectation and goal driven. In a nutshell, you would detect the segments you are looking for.

The difficulties of this approach are threefold. Firstly, the employed analytics – clustering – requires an underlying similarity measure which typically reduces the data to numeric features. Cluster analysis that can work with symbolic attributes do exist, but are less well-known and typically not supported by commercially available software.

The existence of a similarity measure is required, otherwise neither cluster analysis can be applied nor can customers be assigned to clusters.

Secondly, assigning customers to segments is a problem, because survey data that was used to form segments is not available for the vast majority of customers. That means customers are assigned into segments by available information about them which at best contains data about the products and services they use but at worst is based only on rather inadequate data like the postcode, for example.

Thirdly, the segmentation approach is to a large extent goal driven and static. That means the data that is used has been collected with the assumption that it is ultimately relevant for segmentation. Data or attributes not considered to be relevant are dropped from the analytical process early on to make cluster analysis feasible and aid the interpretation of detected clusters which is essential to form meaningful segments. The danger of this approach is that potentially relevant features are ignored meaning certain segments may not be detected. The approach is also static, which means that once segments have been established change in those segments is not monitored because of the practical repercussions like regularly running expensive surveys etc. This results in missing important trends, threats and opportunities because segments and change in several ways. New groups can appear, disappear, merge, move, shrink or grow.

A promising approach would be to concentrate on data that is actually relevant in describing the relationship between customers and the business, i.e. data about interactions with customers and their usage profile of products and services. The data would be a mixture of symbolic data, like product types, fault codes, complaint reasons etc and numeric data on different scales like counts, costs, revenues, frequencies etc. If data types have to be consolidated it is typically better to discretise numerical data and lose some information instead of turning symbolic values into numbers and thus introducing spurious information like distances and relations.

In this paper we are looking at using association rule mining for detecting *interesting* segments in data. We define interesting segments as segments that display some temporal change reflected in the data. We relate growing or shrinking segments to threats and opportunities the business must know about. We explain how tracking the temporal changes of support and confidence values can lead to a notion of interestingness. We will illustrate our approach by applying it to two data sets from customer surveys and network usage.

2 Related Work

Customer segmentation is the process of dividing customers into homogeneous groups on the basis of common attributes. In most application customer segmentation is accomplished by defining numerical attributes which describe a customer's value based on economical and market considerations. Cluster algorithms are then commonly employed in order to discover groups of customers with similar attribute values. For example, in [1] three different clustering algorithms are compared to segment stock trading customers based on their amount of trade in different trading scenarios. Segmentation methods based on clustering require a user to carefully select the used attributes by hand in a tedious process. Since the number of used attributes is rather low, commonly only

two or three, the analysis of segment change can still be done manually. This might be the reason why to our knowledge no automated approach has been published yet.

Several approaches have been proposed to analyse changes in customer behaviour, for instance in retail marketing [2], in an internet shopping mall [3,4] and in an insurance company [5]. These approaches typically compare two sets of rules generated from datasets of two different periods. For rule representation either decision trees [5,4] or association rules [3,2] are used. For example, in a telecommunication retail application, such approaches may detect that customers used to order a certain tariff with a certain special option—now they still order this tariff, but seldom with the special option. The aforementioned approaches only detect *what* has changed rather than *how* something changes. Picking up on the last example this means, they cannot spot the declining trend in the ordered special options. Spotting trends, however, is crucial for many cooperations.

In the area of association rules, the discovery of interesting changes has been studied by several authors. In [6] a query language for shapes of histories is introduced. A fuzzy approach to reveal the regularities in how measures for rules change and to predict future changes was presented by [7]. A framework to monitor the changes in association rule measures based on simple thresholds for support and confidence is described in [8].

3 Preliminaries

3.1 Frequent Itemsets

We define a customer segment as a set of customers which have common features or attributes. Given a data set which describes customers any attribute value combination of each subset of its attributes therefore qualifies as a candidate customer segment. However, we are only interested in customer segments which are frequent in relation to the overall population. This means, we do not aim for segments which present only a tiny fraction of customers, but for those which are larger than an user defined frequency threshold.

Customer segments defined this way can be represented by *frequent itemsets*. The discovery of frequent itemsets is a broadly used approach to perform a nearly exhaustive search for patterns within a data set [9]. Its goal is to detect all those attribute values which occur together within a data set and whose relative frequency exceeds a given threshold. The advantage of frequent itemset discovery is the completeness of its results: it finds the exhaustive set of all significant patterns. For this reason it provides a rather detailed description of a data set's structure. On the other hand, however, the set of discovered itemsets is typically vast.

Formally, frequent itemset discovery is applied to a set \mathcal{D} of *transactions* $\mathcal{T} \in \mathcal{D}$. Every transaction \mathcal{T} is a subset of a set of items \mathcal{L} . A subset $\mathcal{X} \subseteq \mathcal{L}$ is called *itemset*. It is said that a transaction \mathcal{T} *supports* an itemset \mathcal{X} if $\mathcal{X} \subseteq \mathcal{T}$. As usual, the frequency of an itemset \mathcal{X} is measured by its *support* $\text{supp}(\mathcal{X})$ which estimates $P(\mathcal{X} \subseteq \mathcal{T})$, or short $P(\mathcal{X})$. For example, suppose that we are given a data set, which contains survey results about customer satisfaction, the following frequent itemset could have been discovered from it:

AGE > 50 , SATISFIED = YES

The support of this itemset is the relative frequency of customers that are over 50 years old and satisfied, i.e., it describes the relative size of the customer segment.

In the following we will use the notions of a customer segment and a frequent itemsets synonymously.

3.2 Support Histories

The underlying idea of our framework is to detect interesting changes in a customer segment, represented by an itemset, by analysing the support of the itemset along the time axis. The starting point of such an approach is as follows: a timestamped data set is partitioned into intervals along the time axis. Frequent itemset discovery is then applied to each of these subsets. This yields sequences—or *histories*—of support for each itemset, which can be analysed further. Of particular interest are regularities in the histories which we call *change patterns*. They allow us to make statements about the future development of a customer segment and thus provide a basis for proactive decision making.

Let \mathcal{D} be a time-stamped data set and $[t_0, t_n]$ the minimum time span that covers all its tuples. The interval $[t_0, t_n]$ is divided into $n > 1$ non-overlapping periods $T_i := [t_{i-1}, t_i]$, such that the corresponding subsets $\mathcal{D}(T_i) \subset \mathcal{D}$ each have a size $|\mathcal{D}(T_i)| \gg 1$. Let $\hat{T} := \{T_1, \dots, T_n\}$ be the set of all periods, then for each $T_i \in \hat{T}$ frequent itemset discovery is applied to the transaction set $\mathcal{D}(T_i)$ to derive item sets $\mathcal{I}(\mathcal{D}(T_i))$.

Because the support of every itemset \mathcal{X} is now related to a specific transaction set $\mathcal{D}(T_i)$ and thus to a certain time period T_i we need to extend its notation. This is done straightforward and yields $\text{supp}(\mathcal{X}, T_i) \approx P(\mathcal{X} | T_i)$. Each itemset $\mathcal{X} \in \hat{\mathcal{I}}(\mathcal{D}) := \bigcap_{i=1}^n \mathcal{I}(\mathcal{D}(T_i))$ is therefore described by n values for support. Imposed by the order of time the values form a sequence called *support history* $H(\mathcal{X}) := (\text{supp}(\mathcal{X}, T_1), \dots, \text{supp}(\mathcal{X}, T_n))$ of the itemset \mathcal{X} . These histories are then used in subsequent steps to detect interesting change patterns.

To continue our example, suppose that we may discover that the support of the itemset

$$\text{AGE} > 50, \text{SATISFIED} = \text{YES}$$

has an downward trend. This, in turn, can be interpreted as that the group of all satisfied customers over 50 steadily gets smaller.

4 Architecture of the Framework

As already mentioned above our approach builds upon the idea of deriving frequent itemsets as representations of customer segments at different points in time, which are then analysed for changes. To derive a history, data sets collected during many consecutive periods have to be analysed for frequent itemsets. After each analysis session the discovered itemsets have to be compared to those discovered in previous periods and their histories have to be extended. On the other hand, history values may be discarded if their age exceeds an application dependent threshold. Therefore, itemsets and histories have to be stored on a long term basis. Taking all of the aforesaid into account the first task of our framework is:

1. Frequent itemsets have to be *discovered* and their histories efficiently stored, managed and maintained.

If histories with a sufficient length are available, the next task is straightforward:

2. Histories that exhibit specific change patterns have to be reliably *detected*.

Frequent itemset discovery is generally connected with two problems. In the first place, a vast number of itemsets will be detected. Secondly, frequent itemsets may be obvious, already known or not relevant.

Since a history is derived for each rule, the first problem also affects our framework: it has to deal with a vast number of histories and thus it is likely that many change patterns will be detected. Moreover, as we will briefly discuss in Section 5, methods that were developed to deal with this problem for itemsets cannot be used when it comes to analyzing change. Furthermore, there is also a quality problem: not all of the detected change patterns are equally interesting to a user and the most interesting are hidden among many irrelevant ones. Overall, the third task is:

3. Histories with a change pattern have to be analysed for redundancies and *evaluated* according to their interestingness.

Because the aforementioned tasks build upon each other, they can be seen as layers of a processing framework. According to their task the layers are termed *Segment Detector*, *Change Analyser* and *Interestingness Evaluator*, respectively.

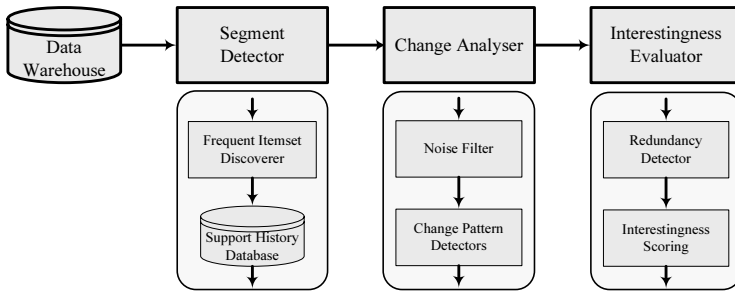


Fig. 1. Architecture of our framework

5 Segment Detector

Given a timestamped data set collected during a certain period, the task of the Segment Detector is to discover and store the customer segments in it. Since in our application each frequent itemset is a potentially interesting customer segment the first component of this layer is an algorithm for frequent itemset discovery, its second component is a database that stores and manages itemsets and their histories. Both components, but also the choice of the time periods, will be explained in the following.

In order to obtain the data set, the period length has to be chosen. Two aspects have to be considered. Long periods lead to many transactions in the individual data sets for

the different periods and thus can enhance the reliability of the calculated support. Short periods allow to measure support more frequently, which may lead to a more reliable and earlier detection of change patterns. The choice of periods length should therefore depend on the available amount of data.

After the data set is available, frequent itemset discovery is applied to it. A typical approach may not only consist of the discovery method itself, but also of methods for pruning and constrained mining. Such methods have been developed to cope with the aforementioned problem of a vast amount of discovered itemsets in each period. This itemset quantity problem directly affects our application. A huge number of histories has to be processed and consequently far too many change patterns will be reported. In order to cope with this problem, pruning methods have been developed in order to constrain the itemsets. From the perspective of our framework such pruning methods treat itemsets generated in different time periods independently from another. However, in our application we process many, temporally ordered itemsets. Thus the itemset property utilized for pruning—in general a measure based on itemset statistics—may vary for some itemsets over time, but still match the pruning criterion in each itemset. Although these variations may render itemsets interesting, they are discarded by existing approaches for itemset pruning. Consequently, we cannot directly use them.

6 Change Analyzer

The task of the *Change Analyzer* is to discover change patterns in support histories. In this paper, however, we only discuss how histories are detected that are stable or exhibit a trend. The Change Analyzer fulfills its task by a two step approach. In the first step a filter is applied to the histories to reduce the noise contained in them. In a second step statistical tests for trend and stability are conducted.

Support histories inherently may contain random noise. Random noise may influence subsequent analysis steps in such a way that wrong and misleading results are produced. To reduce this effect we use *double exponential smoothing* [10] in order to reveal more clearly any trend or stability. It is a simple and fast, yet effective method, which can easily be automated.

A trend is present if a history exhibits steady upward growth or a downward decline over its whole length. This definition is rather loose, but in fact there exists no fully satisfactory definition for trend [10]. From a data mining perspective a trend describes the pattern that each value is likely to be larger or smaller than all its predecessors within a sequence, depending on whether the trend is upward or downward. Hence it is a qualitative statement about the current and likely future development of a history. However, taking aspects of interpretability and usefulness into account, such a statement is sufficient in the case of our application. When faced with a vast number of customer segments and their histories, a user often has a basic expectation whether they should exhibit a trend and of what kind. By comparing his expectations with reality he will mostly be able to roughly assess the implications for his business. On the other hand, a user will rarely know in advance how trends should look like quantitatively, for example, their shape or target values. Thus he may be unable to exploit the advantages of more sophisticated trend descriptions, like regression models.

To choose a method for trend detection, it has to be taken into account that the number of histories to examine is huge. Whenever a trend is reported the user is basically forced to rely on the correctness of this statement, because it is infeasible for him to verify each trend manually. In addition to the requirement of reliable detection, the method should incorporate no assumptions about any underlying model, because it is very unlikely that it will hold for all or at least most sequences. Therefore non-parametric statistical tests are the appropriate choice for trend detection.

Within our framework we provide two statistical tests for trend, the *Mann-Kendall test* [11] and the *Cox-Stuart test* [12]. The Cox-Stuart test exploits fewer features of the history, leading to a computational effort that increases linearly with the history length. Although this may render the Cox-Stuart test susceptible to noise, because the influence of artefacts on the test result is stronger, it is considerably faster for long histories. In contrast to this, the Mann-Kendall test is much more robust, but its computational effort increases quadratically with the history length. Therefore it has to be determined which of the two issues—speed or robustness—is more important depending on the actual characteristics of the data used.

Roughly speaking, a history is considered stable if its mean level and variance are constant over time and the variance is reasonably small. Similar to trends, a clear definition of stability is difficult. For example, a history may exhibit a cyclical variation, but may nevertheless be stable on a long term scale. Depending on the actual interest of a user, either the one or the other may have to be emphasised. From a data mining perspective stability describes the pattern that each value is likely to be close to a constant value, estimated by the mean of its predecessors. Thus it is, like a trend, a qualitative statement about the future development of a history. However, in contrast to a trend, it can easily be modeled in an interpretable and useful way, e.g., by the sample mean and variance. Generally, stable customer segments are more reliable and can be trusted—an eminently useful and desirable property for long term business planning.

To test for stability we use a method based on the well-known χ^2 test. However, since the χ^2 test does not take the inherent order of a history's values into account, our method may infrequently also classify histories as stable, which actually exhibit a trend. Therefore, we chose to perform the stability test as the last one in our sequence of tests for change patterns.

7 Interestingness Evaluator

Since usually a vast number of change patterns for customer segments will be detected, it is essential to provide methods which reduce their number and identify potentially interesting ones. This is the task of the *Interestingness Evaluator*. To reduce the number of candidate segments the Interestingness Evaluator contains a redundancy detection approach, based on so-called derivative histories [13]. Although this approach proves to be very effective, the number of temporally non-redundant customer segments may still be too large for manual examination. Therefore a component for interestingness evaluation is provided, which contains a set of interestingness measures.

7.1 Redundancy Detection

Generally, most changes captured in a segment’s history—and consequently also change patterns—are simply the snowball effect of the changes of other segments. Suppose we are looking at customer satisfaction surveys and our framework would discover that the support of the segment

$$\mathcal{X}_1 : \text{AGE} > 50, \text{SATISFIED}=\text{YES}$$

shows an upward trend. That is, the fraction of customers over 50 who are satisfied increases. However, if the fraction of males among all over 50 year old satisfied customers is stable over time, the history of

$$\mathcal{X}_2 : \text{AGE} > 50, \text{GENDER}=\text{MALE}, \text{SATISFIED}=\text{YES}$$

shows qualitatively the same trend. In fact, the history of segment \mathcal{X}_2 can be *derived* from the one of \mathcal{X}_1 by multiplying it with a gender related constant factor. For this reason, the segment \mathcal{X}_2 is *temporally redundant* with respect to its support history.

It is reasonable to assume that a user will generally be interested in customer segments with non-derivative and thus non-redundant histories, because they are likely key drivers for changes. Moreover, derivative segments may lead to wrong business decisions. In the above example a decision based on the change in segment \mathcal{X}_2 would account for the gender as one significant factor for the observed trend. In fact, the gender is completely irrelevant. Therefore, the aim is to find segments that are non-redundant in the sense that their history is not a derivative of related segments’ histories. In a way, the approach is searching and discarding segments that are not the root cause of a change pattern which, in turn, can be seen as a form of pruning. In order to find derivative segments we have to answer the following questions. First, what is meant by *related* itemsets (segments, respectively), and second, what makes a history a *derivative* of other histories. Regarding the first question, we use the superset relation to define *related itemsets*: an itemset \mathcal{Y} is related to an itemset \mathcal{X} iff $\mathcal{X} \prec \mathcal{Y} := \mathcal{X} \supset \mathcal{Y}$. We also say that \mathcal{Y} is *more general* than \mathcal{X} because its supporting transaction set is larger. In the following we write $\mathcal{X}\mathcal{Y}$ for $\mathcal{X} \cup \mathcal{Y}$. We then define:

Definition 1. Let $\mathcal{X}, \mathcal{X}_1, \mathcal{X}_2 \dots \mathcal{X}_p$ be itemsets with $\mathcal{X} \prec \mathcal{X}_i$ for all i and $p > 0$. Let the \mathcal{X}_i be pairwise disjoint. Let supp the support, $\text{supp}(T) := \text{supp}(\mathcal{X}, T)$ and $\text{supp}_i(T) := \text{supp}(\mathcal{X}_i, T)$ its functions over time and $\mathcal{M} := \{g : \mathbb{R} \rightarrow \mathbb{R}\}$ be the set of real-valued functions over time. The history $H(\mathcal{X})$ is called derivative iff a function $f : \mathcal{M}^p \rightarrow \mathcal{M}$ exists such that for all $T \in \hat{T}$

$$\text{supp}(T) = f(\text{supp}_1, \text{supp}_2, \dots, \text{supp}_p)(T) \tag{1}$$

For simplicity, we call an itemset *derivative* iff its history is derivative.

The main idea behind the above definition is that the history of an itemset is derivative, if it can be constructed as a mapping of the histories of more general itemsets. To compute the value $\text{supp}(\mathcal{X}, T)$ the values $\text{supp}(\mathcal{X}_i, T)$ are thereby considered. The definition above does not allow for a pointwise definition of f on just the $T \in \hat{T}$, but

instead states a general relationship between the support values independent from the point in time. It can therefore be used to predict the value of $\text{supp}(\mathcal{X})$ given future values of the $\text{supp}(\mathcal{X}_i)$. A simple example we will see below is $\text{supp} = f(\text{supp}_1) = c \cdot \text{supp}_1$, i.e. the support history can be obtained by multiplying the support history of a more general itemset with a constant c .

In the following we introduce two criteria for detecting derivative support histories which can be used in combination or independently from another. The functions f are quite simple and we make sure that they are intuitive.

The first criterion checks if the support of an itemset can be explained with the support of exactly one less specific itemset.

Criterion 1. *The term $\text{supp}(\mathcal{X}\mathcal{Y}, T) / \text{supp}(\mathcal{Y}, T)$ is constant over $T \in \hat{T}$ given disjoint itemsets \mathcal{X} and \mathcal{Y} .*

The meaning of the criterion becomes clear when being rewritten as

$$c = \text{supp}(\mathcal{X}\mathcal{Y}, T) / \text{supp}(\mathcal{Y}, T) = P(\mathcal{X}\mathcal{Y} | T) / P(\mathcal{Y} | T) = P(\mathcal{X} | \mathcal{Y}T)$$

with a constant c . The probability of \mathcal{X} is required to be constant over time given \mathcal{Y} , so the fraction of transactions containing \mathcal{X} additionally to \mathcal{Y} constantly grows in the same proportion as \mathcal{Y} . For this reason the influence of \mathcal{X} in the itemset $\mathcal{X}\mathcal{Y}$ on the support history is not important. Due to

$$\text{supp}(\mathcal{X}\mathcal{Y}, T) = c \cdot \text{supp}(\mathcal{Y}, T) \quad (2)$$

with $c = \text{supp}(\mathcal{X}\mathcal{Y}, T) / \text{supp}(\mathcal{Y}, T)$ for any $T \in \hat{T}$, $\mathcal{X}\mathcal{Y}$ is obviously a derivative of \mathcal{Y} with respect to support history as defined in Definition 1.

Figures 2 and 3 show an example of a derivative support history. Figure 2 shows the support histories of the less specific itemset at the top and the more specific itemset underneath over 20 time periods. The shape of the two curves is obviously very similar and it turns out that the history of the more specific rule can be approximately reconstructed using the less specific one based on (2). As shown in Figure 3, the reconstruction is not exact due to noise. A suitable statistical test was proposed in [13].

Opposed to the criterion above, the following is based on the idea of explaining the support of an itemset with the support values of two subsets.

Criterion 2. *The term $\frac{\text{supp}(\mathcal{X}\mathcal{Y}, T)}{\text{supp}(\mathcal{X}, T) \text{supp}(\mathcal{Y}, T)}$ is constant over $T \in \hat{T}$ given disjoint itemsets \mathcal{X} and \mathcal{Y} .*

$\text{supp}(\mathcal{X}\mathcal{Y}, T)$ measures the probability of the itemset $\mathcal{X}\mathcal{Y}$ in period T which is $P(\mathcal{X}\mathcal{Y} | T)$. The term $\frac{\text{supp}(\mathcal{X}\mathcal{Y}, T)}{\text{supp}(\mathcal{X}, T) \text{supp}(\mathcal{Y}, T)} = \frac{P(\mathcal{X}\mathcal{Y} | T)}{P(\mathcal{X} | T)P(\mathcal{Y} | T)}$ is quite extensively used in data mining to measure the degree of dependence of \mathcal{X} and \mathcal{Y} at time T . The criterion therefore expresses that the degree of dependence between both itemsets is constant over time.

The support history of $\mathcal{X}\mathcal{Y}$ can then be constructed using

$$\text{supp}(\mathcal{X}\mathcal{Y}, T) = c \cdot \text{supp}(\mathcal{X}, T) \text{supp}(\mathcal{Y}, T) \quad (3)$$

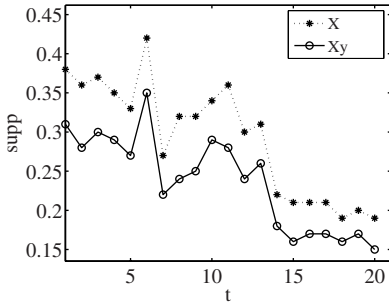


Fig. 2. Histories of the segment \mathcal{X} and its derivative segment $\mathcal{X}y$

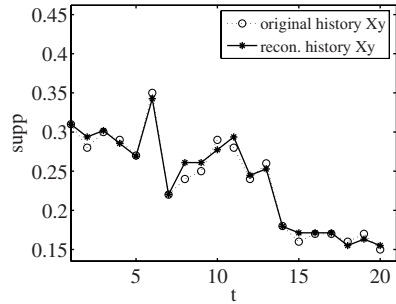


Fig. 3. Reconstructed history of $\mathcal{X}y$ using the history of \mathcal{X}

with $c = \text{supp}(\mathcal{X}\mathcal{Y}, T) / (\text{supp}(\mathcal{X}, T) \text{supp}(\mathcal{Y}, T))$ for any $T \in \hat{T}$, that is the individual support values of the less specific itemsets are used corrected with the constant degree of dependence on another. According to Definition 1 the support history of $\mathcal{X}\mathcal{Y}$ is therefore derivative.

Overall, an itemset is considered derivative if more general itemsets can be found, such that at least one of the Criteria 1 or 2 holds.

7.2 Interestingness Scoring

To assess the interestingness of detected trends and stabilities it has to be considered that each history is linked to a segment which itself has a certain relevance to a user. The detection of a specific change pattern may significantly influence this prior relevance. However, there is no broadly accepted and reliable way of measuring an itemset’s interestingness up to now [14]. Therefore we consider any statement about the interestingness of a history also as a statement about the interestingness of its related itemset.

To assess stable histories two things should be considered: in the first place, most data mining methods typically assume that the domain under consideration is stable over time. Secondly, support is an interestingness measure for itemsets themselves. Taking all this into account, a stable history is in some way consistent with the abovementioned assumption of data mining. It is summarised by the mean of its values, which in turn can then be treated as an objective interestingness measure. Here the variance of the history can be neglected, since it is constrained by the stability detection method.

To develop objective interestingness measures for trends is more complex due to their richness of features. For identifying salient features of a given trend, it is essential to provide reference points for comparison. As such we chose the assumptions a user naively makes in the absence of any knowledge about the changes in support histories. From a psychological perspective they can be seen as the anchors relative to which histories with a trend are assessed: a trend becomes more interesting with increasing inconsistency between its features and the user’s naive assumptions. We identified three

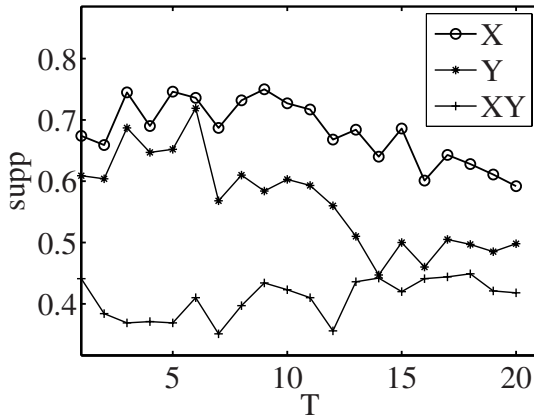


Fig. 4. Examples of interesting histories which exhibit a trend

such assumptions and defined heuristic measures for the discrepancy between a history and an assumption:

- **Stability:** Unless other information is provided, a user assumes that histories are stable over time. This assumption does not mean that he expects no trends at all, but expresses his naive expectations in the absence of precise knowledge about a trend. It should be noted that this is consistent with many data mining approaches, which implicitly assumes that the patterns hidden in the data are stable over time. The histories of the segment \mathcal{XY} in Figure 4 would violate the stability assumption because its trend is very clear.
- **Non-rapid Change:** Since a user shapes its business, he will be aware that the domain under consideration changes over time. However, he will assume that any change is continuous in its direction and moderate in its value. For example, if a business starts a new campaign, it will probably assume that the desired effect on the customers evolves moderately, because, for instance, not all people will see a commercial immediately. On the other hand, a rapid change in this context attracts more attention, because it may hint at an overwhelming success or an undesired side effect. For example, the history of the segment \mathcal{Y} in Figure 4 would be very interesting according to the non-rapid change assumption because the depicted trend is very pronounced and steep.
- **Homogeneous Change:** If the support of an itemset changes over time, it is assumed that the rate and direction of changes in the support of all its supersets are the same. This basically means that the observed change in the itemset does not depend on further items. For example, a user may know that the fraction of satisfied customers increases. The homogeneous change assumptions states that the observed change in satisfaction affects all customers and not only selected sub-populations, e.g. customers over fifty. For example, the fraction of satisfied males among all customers may increase. According to the homogeneous change assumption a user would conclude that the fraction of all satisfied married male customers increases at the same rate. For example, the history of the segments \mathcal{XY} in Figure 4

would be very interesting because its shape is completely different from those of its more general segments.

8 Experimental Evaluation

To evaluate our framework we chose two representative real-life dataset. One contains answers of residential customers to a survey collected over a period of 40 weeks. The other contains network usage data of business customers collected over a period of 9 months. We transformed each dataset into a transaction set by recoding every (attribute, attribute value) combination as an item.

In the survey dataset each tuple is described by 19 nominal attributes with a domain size between 2 and 10. We split the transaction set into 20 subsets, each corresponding to a period of two weeks. The subsets contain between 829 and 1049 transactions. From each subset we derived frequent itemsets (customer segments, respectively) with a support greater than 0.04 and not more than 5 describing attributes per segment. From the obtained 20 frequent itemsets we created a compound itemset by intersecting them. Its size is 1202.

The network usage dataset is described by 24 nominal attributes with a domain size of 5. We split the transaction set into 9 subsets each covering a period of one month and having a size of 37 transactions. From each subset we derived frequent itemsets (customer segments, respectively) with a support greater than 0.1 and not more than 5 describing attributes per segment. The intersection of these itemset has a size of 8984.

Subsequently we applied the proposed framework using the Mann-Kendall test for trend detection. Thereby two objectives are linked with our evaluation. First, the number of segments exhibiting trends or stabilities has to be determined. Second, the number of derivative rule histories has to be determined. The results of our analysis are shown in Table 1 and Table 2.

Table 1. Absolute number of segments which exhibit a trend or are stable differentiated by non-redundancy

	segments		downward trend		upward trend		stable	
	all	non-redund.	all	non-redund.	all	non-redund.	all	non-redund.
Surveys	1202	457	50	31	147	50	830	307
Network Usage	8984	1909	3030	294	100	43	5854	1572

As we can see the number of segments which exhibit a trend or stability strongly depends on the data set. For example, in the survey data set approximately 4.2% of the segments show an upward trend compared to 33.7% for the network usage data. It shows, however, that segments which exhibit some kind of regular change exist and that they can be rather frequent. Looking in the columns for non-redundant changes we can see that only a small fraction of changing segments cannot be explained by the change of more general segments. As we discussed earlier, segments with redundant changes can lead to suboptimal business decisions. As we see in our results they also

Table 2. Relative number of segments which exhibit a trend or are stable differentiated by non-redundancy

	segments		downward trend		upward trend		stable	
	all	non-redund.	all	non-redund.	all	non-redund.	all	non-redund.
Surveys	100.0%	38.0%	4.2%	2.6%	12.2%	4.2%	69.1%	25.5%
Network Usage	100.0%	21.2%	33.7%	3.3%	1.1%	0.5%	65.2%	7.5%

significantly increase the number of changing segments. This, again, underlines the need for redundancy detection in our framework for which we provided a powerful method.

9 Conclusions

We have shown how association rule mining, combined with tracking temporal developments of support and confidence and the application of an interestingness notion can be used for detecting and monitoring customer segments. This is a very important challenge for customer-focussed enterprises. Many businesses regularly collect huge volumes of time-stamped data about all kinds of customer interactions. This data reflects changes in customer behavior. It is crucial for the success of most businesses to detect these changes, correctly interpret their causes and finally to adapt or react to them. Hence there is a significant need for data mining approaches that are capable of finding the most relevant and interesting changes in a data set.

We have proposed a framework for our approach that can provide detailed knowledge about how customer behaviour evolves over time. We successfully applied our framework to two problem domains which are very significant for a telecommunications company: customer analytics, to investigate what is likely to drive customer satisfaction in the future, and network usage, to understand the drivers of change in customer behavior when they are using services.

References

1. Shin, H., Sohn, S.: Segmentation of stock trading customers according to potential value. *Expert Systems with Applications* 27(1), 27–33 (2004)
2. Chen, M.C., Chiu, A.L., Chang, H.H.: Mining changes in customer behavior in retail marketing. *Expert Systems with Applications* 28(4), 773–781 (2005)
3. Song, H.S., Kim, J.K.: Mining the change of customer behavior in an internet shopping mall. *Expert Systems with Applications* 21(3), 157–168 (2001)
4. Kim, J.K., Song, H.S., Kim, T.S., Kim, H.K.: Detecting the change of customer behavior based on decision tree analysis. *Expert Systems* 22(4), 193–205 (2005)
5. Liu, B., Hsu, W., Han, H.S., Xia, Y.: Mining changes for real-life applications. In: Kambayashi, Y., Mohania, M.K., Tjoa, A.M. (eds.) *DaWaK 2000*. LNCS, vol. 1874, pp. 337–346. Springer, Heidelberg (2000)
6. Agrawal, R., Psaila, G.: Active data mining. In: *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining*, pp. 3–8 (1995)

7. Au, W.H., Chan, K.: Mining changes in association rules: a fuzzy approach. *Fuzzy Sets and Systems* 149(1), 87–104 (2005)
8. Spiliopoulou, M., Baron, S., Günther, O.: Efficient monitoring of patterns in data mining environments. In: Kalinichenko, L.A., Manthey, R., Thalheim, B., Wloka, U. (eds.) *ADBIS 2003*. LNCS, vol. 2798, pp. 253–265. Springer, Heidelberg (2003)
9. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Washington DC, pp. 207–216. ACM Press, New York (1993)
10. Chatfield, C.: *Time-Series Forecasting*. Chapman and Hall/CRC, New York (2001)
11. Mann, H.: Nonparametric tests against trend. *Econometrica* 13, 245–259 (1945)
12. Cox, D., Stuart, A.: Some quick sign tests for trend in location and dispersion. *Biometrika* 42, 80–95 (1955)
13. Böttcher, M., Spott, M., Nauck, D.: Detecting temporally redundant association rules. In: *Proceedings of 4th International Conference on Machine Learning and Applications*, pp. 397–403. IEEE Computer Society Press, Los Alamitos (2005)
14. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right objective measure for association analysis. *Information Systems* 29(4), 293–313 (2004)

Collaborative Filtering Using Electrical Resistance Network Models

Jérôme Kunegis and Stephan Schmidt

DAI-Labor, Technische Universität Berlin, Franklinstraße 28, 10587 Berlin, Germany
{jerome.kunegis, stephan.schmidt}@dai-labor.de

Abstract. In a recommender system where users rate items we predict the rating of items users have not rated. We define a rating graph containing users and items as vertices and ratings as weighted edges. We extend the work of [1] that uses the *resistance distance* on the bipartite rating graph incorporating negative edge weights into the calculation of the resistance distance. This algorithm is then compared to other rating prediction algorithms using data from two rating corpora.

1 Introduction

In recommender systems items are recommended to users. Collaborative filtering is a common approach to implementing recommender systems. One way of implementing a recommender system is by collaborative filtering. Instead of calculating an item score based on item features, a collaborative filtering algorithm analyzes existing ratings from users to predict the rating of an item a certain user has not seen.

Items are typically documents, songs, movies or anything that can be recommended to users and that users can rate. Ratings can be given by users explicitly such as with the five star scale used by some websites or can be collected implicitly by monitoring the users' actions such as recording the number of times a user has listened to a song.

The approach presented here models rating databases as bipartite graphs with users and items as the two vertex sets and ratings as weighted edges. [1] describes how this graph can be seen as a network of electrical resistances and how the total equivalent resistance between any two nodes can be used to define a similarity function either between two users, two items or users and items. In the referenced paper edges are not weighted and thus all resistances are modeled as unit resistances.

We extend this work by using the actual ratings as edge weights. Because ratings can be negative, modifications to the previous approach are necessary in order for the similarity function we define to satisfy three basic conditions, which we present in the section defining the rating graph. We use this similarity for predicting ratings and compare the results to common prediction algorithms.

Based on the graph representation of a rating database we can formulate three conditions a good prediction measure should satisfy:

Parallelity. Parallel paths of edges between two nodes contribute monotonically to the similarity between the two nodes. If multiple paths exist between two nodes, the overall similarity between the two nodes should be greater than the similarity taken on each path separately.

Transitivity. A long path of edges with positive weight results in a lower similarity than a short path with similar weights.

Negative ratings. A path consisting of both positive and negative edges leads to a negative similarity exactly if the number of negative edges is odd. This follows from the observation that users that both like or both dislike an item are similar while two users are not similar when one of them like an item the other one dislikes.

We will show later that our method satisfies all three conditions while the original algorithm [1] only satisfies the first two.

Outline. In Section 2 we present mathematical definitions and the two basic rating prediction methods: rating normalization and weighted rating sum based on the Pearson correlation between users. Section 3 defines the bipartite rating graph and the meaning of the rating values. Section 4 defines the similarity between two graph nodes based on electrical resistances in the case that no ratings are negative. In Section 5, the usage of negative ratings is explored and a modified formula is presented that satisfies the three conditions presented above. Section 6 discusses algorithms for solving the system of equations. In Section 7 our similarity measure is compared to two standard prediction measures and to the prediction measure defined in [1]. Section 8 finishes with some remarks about the accuracy of our method and with future work.

2 Related Work

This section presents the two prediction methods used in our evaluation that are not based on the rating graph. The method of [1] is presented after the definition of the rating graph.

2.1 Definitions

Let $\mathcal{U} = \{U_1, U_2, \dots, U_m\}$ be the set of users and $\mathcal{I} = \{I_1, I_2, \dots, I_n\}$ the set of items.

The rating by user U_i of item I_j is denoted r_{ij} . We keep r_{ij} undefined if user U_i has not rated item I_j .

Let \mathcal{I}_i be the set of items rated by user U_i and \mathcal{U}_j the set of users that rated item I_j .

2.2 Rating Scales

Different rating systems use different rating scales. When user ratings correspond to labels such as *good* or *bad*, they are represented by an integer value ranging for instance from 1 to 5, where 5 represents the highest rating (*very good*), and 1 represents the lowest rating (*very bad*). Users can give neutral ratings when the number of different values is odd. These ratings can be adjusted such that a neutral value is represented by the value 0. They can also be scaled to fit in the $[-1, +1]$ range, but this is not necessary because the data is normalized as described below as a first step in most algorithms. In some cases ratings may be unbounded, for instance if they represent a document view count.

2.3 Normalization

Different users have a different idea of what *good* and *bad* ratings mean, so some users give the extremal ratings rarely while others always rate items as either very good or very bad. Assuming that the distribution of item ratings should be similar for all users we can scale each user's ratings around the mean rating such that the standard deviation of each user's ratings is a constant. Using the same argument, we can offset each user's ratings so that his mean rating is zero. This kind of transformation is described in [2].

2.4 Mean Rating

As the simplest prediction algorithm, we chose the user's mean rating as a prediction for his rating of any item he has not rated. This is equivalent to always predicting a rating of zero after normalization. For each user U_i , we define the mean rating \bar{r}_i :

$$\bar{r}_i = \frac{1}{|\mathcal{I}_i|} \sum_{j \in \mathcal{I}_i} r_{ij}$$

This method is very simple as it does not take into account other users' ratings.

2.5 Pearson Correlation

To take other users' ratings into account, a simple method to predict the rating for a user $U \in \mathcal{U}$ is to calculate the mean rating other users have given to the item in question. However, other users may have a different taste than user U . Therefore we need a similarity function applicable between user U and other users. This similarity value can then be used as weights to get a weighted mean as a rating prediction. The similarity function used can even admit negative values, indicating that users are likely to have opposing tastes.

A similarity function satisfying these conditions is the Pearson correlation between users calculated using normalized ratings as described in [3]. This correlation is normally calculated using only items the two users have both rated. As described in [3], it is also possible to take the correlation over all items rated by at least one user, filling missing ratings with a default value.

Let $\mathcal{I}_{ab} = \mathcal{I}_a \cap \mathcal{I}_b$. The Pearson correlation between the users U_a and U_b is defined as

$$w(a, b) = \frac{\sum_{j \in \mathcal{I}_{ab}} (r_{aj} - \bar{r}_a)(r_{bj} - \bar{r}_b)}{\sqrt{\sum_{j \in \mathcal{I}_{ab}} (r_{aj} - \bar{r}_a)^2 \sum_{j \in \mathcal{I}_{ab}} (r_{bj} - \bar{r}_b)^2}}$$

where \bar{r}_a and \bar{r}_b are taken over \mathcal{I}_{ab} . The rating prediction of item U_j for user U_i is now given by

$$r_{ij}^p = \left(\sum_a w(i, a) \right)^{-1} \sum_a w(i, a) r_{aj} \quad (1)$$

where sums are taken over all users that have rated items in common with user U_i . This expression is not defined when the sum of correlations is zero.

Alternatively, we can use all items at least one user has rated, and substituting these missing ratings with a neutral value such as the user’s mean rating. Throughout this paper only the first option will be used as it is common practice in collaborative filtering recommender systems.

3 The Rating Graph

In this section, we define the weighted rating graph. Given sets of users, items and ratings, users and items are represented by vertices, while ratings are modeled as the edges connecting them. Rating values become edge weights.

As observed by [4] and [5], this rating graph is bipartite. We can assume that the graph is connected. If it is not, it can be made connected by only keeping the connected component containing the vertices we want to compare. If the vertices to be compared are not connected, then the graph cannot be used to predict the rating in question since the vertices are not part of the same rating subgraph. In fact, the absence of a path between two nodes means that there is no way to say anything about the relation between the two users or items, and no algorithm or model can give results in this case.

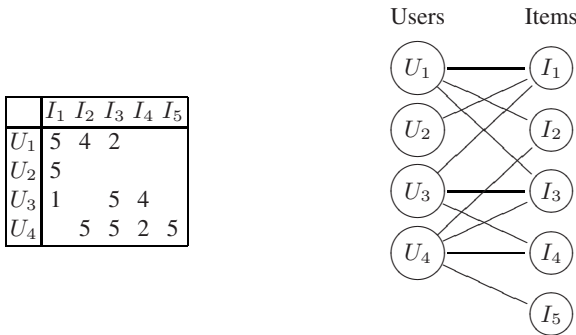


Fig. 1. User-item rating table and graph, on a scale from 1 (very bad) to 5 (very good)

In Figure 1 four users (U_1-U_4) have rated five items (I_1-I_5). However, not all possible ratings were given. Empty cells in the rating table on the left denote ratings users have not given.

The corresponding bipartite rating graph is shown on the right. It contains one edge for each rating. In this case, the graph is connected.

As explained in the introduction, the purpose of the graph is to find connections between users and items. As an example for the graph given above, we could ask the question: How would user A rate item I_5 ? Since the user has not rated the item, we must use the known ratings to predict a rating for the corresponding vertex pair.

Because users and items are both represented as vertices, our method for calculating a similarity them can also be used for calculating similarities of two users or two items. In this paper, we restrict ourselves to calculating the similarity between a user and an item.

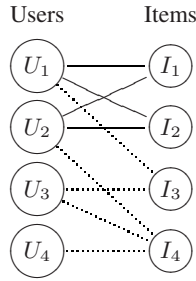


Fig. 2. In this example, the edges not taken into account by the Pearson correlation calculation are shown as dotted lines

In the case covered in [11], the graph is not bipartite, but tripartite: Vertices represent users, movies that can be rated, and categories of movies (such as comedies). In the resulting tripartite graph however, the problems given in [11] can all be modeled as the similarity between two vertices. Therefore, we will consider the general case of graphs that need not necessarily be bipartite.

Given a weighted graph and two vertices, a similarity measure between the two vertices can now be defined. As mentioned before, edges are weighted, and their weights are signed. Positive values indicate a positive association between the nodes, negative values indicate a negative association between the nodes. The measure we want to define must satisfy the three conditions presented in the introduction. These conditions translate to the following mathematical properties:

Long paths. Paths from one vertex to another can be compared by the number of edges they contain. Long paths must correspond to smaller similarities than short paths. If, for instance, two users have rated the same item, then their similarity must be higher than if they had only rated items in common with a third user. However, long paths must not give a similarity of zero. This condition may be called transitivity.

Parallel paths. If a large number of distinct paths exist between two vertices, the similarity value between the two vertices must be higher than for the case where only few paths exist.

Negative edges. If the two vertices are connected by exactly one path, then the resulting similarity must be of the same sign as the product of the weights of the path edges. This condition corresponds to the sensible assumption that one will dislike items that are disliked by users with similar taste.

We will now show that the Pearson correlation based rating prediction does not satisfy all three conditions.

In the bipartite rating graph, the Pearson correlation can only be calculated between two user vertices that are connected by a path of length two. If the distance between the two vertices is longer than two edges, the Pearson correlation cannot be calculated because the users have not rated any items in common. Thus, the Pearson correlation does not satisfy the requirement on long paths. However, it does satisfy the requirement on parallel paths and the requirement on negative edges.

In Figure 2, the Pearson correlation between users U_1 and U_2 is calculated. Edges that are not taken into account during calculation are shown as dotted lines. This example shows that while both user U_1 and user U_2 have something in common with user U_3 , these connections are simply ignored by the Pearson correlation. Furthermore, user U_4 has no ratings in common with user U_1 , so the Pearson correlation cannot be calculated between these two users.

4 Resistance Distance

This section describes the similarity function based on electrical resistance as described in [1].

We have previously proposed that a sensible similarity measure needs to be smaller as paths get longer and larger when there are parallel paths. A similar behavior can be encountered in electrical engineering regarding electrical resistances: When in series, their values add to each other; when in parallel, the inverse of their values add to each other, and the resulting resistance is lower than individual resistances.

Our function is required to yield the opposite result: A path of edges in the rating graph (corresponds to resistances in series) must result in a lower value. Similarly, parallel paths must lead to a rating value that is the sum of the individual path values. Therefore, we use the inverse of the resistance in our function. The inverse of the electrical resistance is the electrical conductance. In this paper, we use the letter r to denote ratings, which must not be confused with the letter R that usually denotes resistances.

In [1] all ratings are initialized with the unit rating ($r_{\text{unit}} = 1$) to avoid negative ratings, which are problematic as we will see below. Figure 3 shows some examples containing only unit resistances (or, equivalently, unit conductances). The corresponding resulting conductance is given for each graph, illustrating how the length of paths influences the similarity value, and how parallel edges result in higher similarity values. As mentioned in [6], the resistance distance between nodes of a graph is a metric.

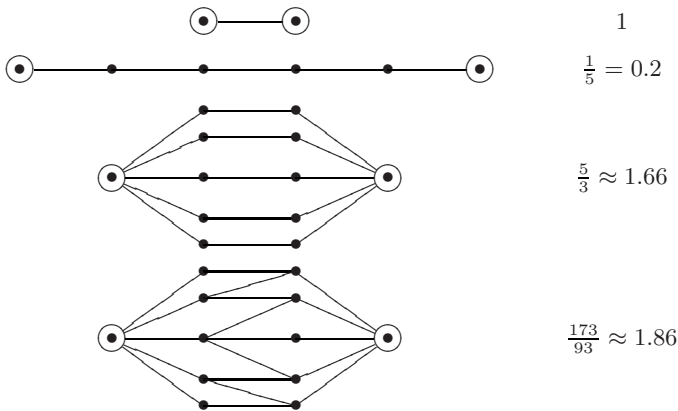


Fig. 3. Bipartite rating graphs annotated with resistance distance values between pairs of highlighted nodes. All edges have unit weight.

We will now describe a method for calculating this total conductance between vertices A and B . When applying a unit voltage on the two vertices, the current through the network will equal the total conductance. In order to compute the value of the current passing through the network, we first need to calculate the potential on A and on the vertices adjacent to A . To this end, we introduce a variable x_V for each vertex V . We then simulate the application of a unit voltage between A and B by setting $x_A = 0$ and $x_B = 1$. For each vertex V adjacent to the vertices $V_1 \dots V_k$, we know that the total current entering V must be zero. Let I_i be the current going from V_i to V and r_i the conductance of the edge (V, V_i) . We obtain the following system of equations:

$$\begin{aligned} \sum_i I_i &= 0 \\ \sum_i r_i(x_V - x_{V_i}) &= 0 \\ \sum_i r_i x_V - r_i x_{V_i} &= 0 \\ \left(\sum_i r_i \right) x_V &= \sum_i r_i x_{V_i} \end{aligned} \quad (2)$$

At this point, an additional variable is introduced for each vertex in the graph. Solving this system of equations will yield potentials for all vertices of the graph. The current flow from A to B is now given by taking the sum of the current over all edges (A, V_i) incident to A :

$$\begin{aligned} I_{AB} &= \sum_i I_{AV_i} \\ &= \sum_i r_{AV_i}(x_{V_i} - x_A) \\ &= \sum_i r_{AV_i} x_{V_i} \end{aligned}$$

The resistance distance is now given by:

$$\begin{aligned} r_{\text{eq}} &= \frac{I_{AB}}{x_B - x_A} \\ &= I_{AB} \\ &= \sum_i r_{AV_i} x_{V_i} \end{aligned} \quad (3)$$

And inverse resistance distance gives our similarity between two nodes A and B :

$$\text{sim}_{\text{unit}} = r_{\text{eq}}^{-1}$$

We observe that finding the total conductance between A and B is equivalent to solving a linear system of n equations and n variables, where n is the number of vertices in the rating graph.

5 Negative Ratings

As we have seen, the inverse resistance distance satisfies the first two conditions we imposed on similarity functions. However, it is easy to see that it does not conform to the third condition; for instance in a path containing an odd number of negative edges, the similarity should be negative, but the inverse resistance distance is always positive.

Instead of just setting all ratings to the value one to avoid negative edge weights, we will try to keep the real values, and try to define a new similarity measure satisfying all three conditions.

We now examine the consequences of inserting the *original*, possibly negative rating values into the equations above. Figure 4 shows a very simple electrical network with negative resistances, and its corresponding inverse resistance distance is calculated using the formula known from physics stating that two resistance in series of magnitude r_1 and r_2 are equivalent to a single resistance of magnitude $\frac{r_1 r_2}{r_1 + r_2}$.

$$\bullet \xrightarrow{r_1 = +1} \bullet \xrightarrow{r_2 = -1} \bullet \quad r_{\text{eq}} = \frac{r_1 r_2}{r_1 + r_2} = \frac{1 \cdot (-1)}{1 + (-1)} = \frac{-1}{0}$$

Fig. 4. The inverse resistance distance on graph with negative edge weights may not be defined

As can be seen, the formula leads to a division by zero. However, we require the result value to be smaller than 1 in absolute value (from the first condition) and negative (from the third condition). The desired result is obtained in a different way; the absolute value of our result is equal to the value calculated using the formula containing the absolute values of the conductances. The sign of the result then has the sign of the product of the two ratings. Thus we want:

$$r_{\text{eq}} = \text{sgn}(r_1) \text{sgn}(r_2) \frac{|r_1| \cdot |r_2|}{|r_1| + |r_2|} = \frac{r_1 \cdot r_2}{|r_1| + |r_2|}$$

Figure 5 displays some examples and their resulting similarities using this formula.

Now, the question arises how the general equations given in Equation 2 are to be adapted to yield the desired result for our simple example cases. If we interpret an edge with rating $-r$ not as a negative resistance of conductance $-r$, but as a resistance with conductance r that “inverts” the potential difference across the resistance, then on the right side of our equation the factor of x_{V_i} remains r_i , but in the sum on the left side the absolute value of r_i must be used. This means that for each vertex V , we replace the equation

$$\left(\sum_i r_i \right) x_V = \sum_i r_i x_{V_i}$$

with

$$\left(\sum_i |r_i| \right) x_V = \sum_i r_i x_{V_i}$$

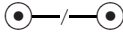
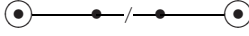
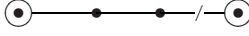
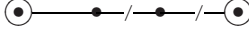
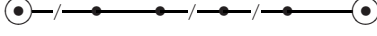
Rating Graph	Similarity
	-1
	-1/3
	-1/3
	1/3
	-1/5

Fig. 5. Rating graphs with negative edges and the similarity between the highlighted nodes calculated taking into account the negative values. Edges with a slash have weight -1 , other edges have weight $+1$.

For the examples of Figure 5 these equations yield the desired results. In the next section, methods for solving the resulting system of equations are given.

6 Y-Δ Simplification

Calculating the resistance distance between two nodes in a network involves solving a system of n equation of n variables, where n is the number of nodes in the network. Before solving a system of equations however, we can try to simplify the network without changing the resistance distance. The examples in Figure 6 show simple cases where this is possible.

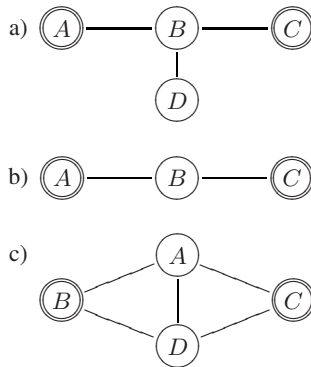


Fig. 6. Simple cases of vertex elimination. The similarity is to be calculated between the doubly circled nodes.

In case a), node D has degree 2. No current can enter this node, so nodes D and B always have the same potential. Therefore, the conductance r_{ab} of the edge (B, D)

has no influence on the resulting conductance. We can just remove this edge without changing the resulting conductance.

In case b), node B has degree 2. This node and its two incident edges can be replaced with a single edge whose resistance is the sum of the two original edges. In this case, the resulting edge weight may become greater than 1 in absolute value.

In the cases a) and b), we have simplified the graph by removing one vertex, removing its incident edges, and in case b) adding one additional edge. In case c) however, the two vertices that could be removed have degree three, so we cannot apply one of the two simplifications as before. Instead, we can use the so-called Y- Δ simplification. We replace node B and its three incident edges (that form a Y) with a triangle of three edges (that form a Δ). Figure 7 shows the Y- Δ transformation graphically. The formulas giving the new resistance values are fundamental to the theory of electrical resistances, as given e.g. in [7].

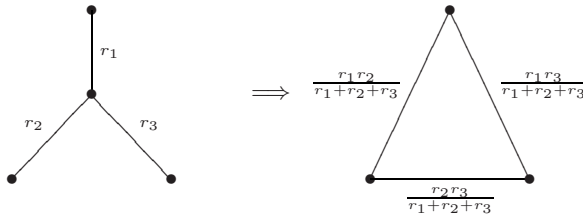


Fig. 7. Principle of Y- Δ simplification

Note that if an edge of the triangle already exists in the graph prior to Y- Δ transformation, the resulting graph will have parallel edges. As parallel conductances are additive, we can just add the new edge value to an existing edge value if necessary.

As further shown in [7], Y- Δ simplification can be generalized to removal of vertices of any degree. In the general case, vertex A of degree k adjacent to vertices $\{B_1, \dots, B_k\}$ with resistance r_i between A and B_i are replaced by $\binom{k}{2}$ new edges. Between each pair of vertices (B_i, B_j) , a new edge is added with weight $\frac{r_i r_j}{\sum_k r_k}$. Figure 8 shows an example for $k = 4$.

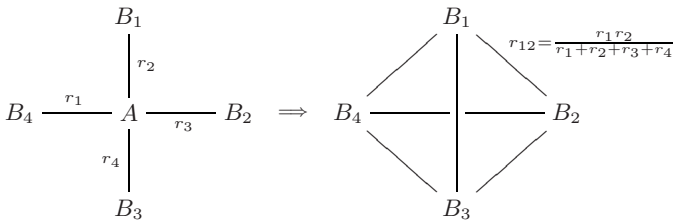


Fig. 8. Elimination of a vertex with degree four

While this generalized removal of nodes works in the simple case of only positive conductance values, it does not work with our modified total conductance. Figure 9 gives an example where according to our definition, the conductance between A and B is $1/5$, but by using simplification of nodes, we get the result $1/3$.

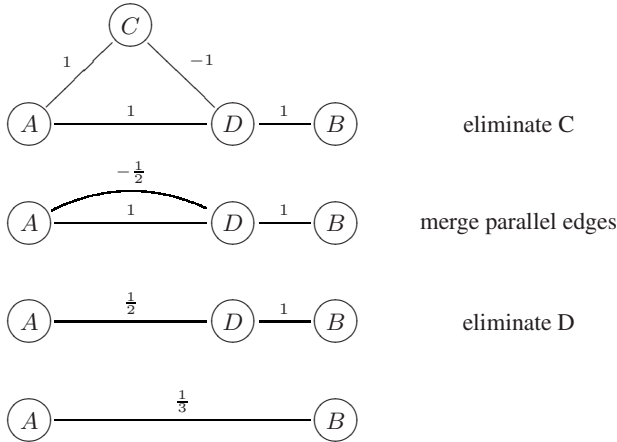


Fig. 9. Simple vertex elimination results in different similarity value than defined

This result, however, does not correspond to the system of equations presented above. The following system of equations must be solved to get the value of the total conductance between A and B according to our definition:

$$\begin{aligned}x_A &= 0 \\x_B &= 1 \\2x_C &= x_A - x_D \\3x_D &= x_A + x_B - x_C\end{aligned}$$

Solving this system, we get $r_{eq} = x_C + x_D = 1/5$. This example shows that Y- Δ elimination can not be applied in the case of modified resistance distance calculation. Therefore, the system of equations defined by Equation 2 must be solved. As shown in [8] and [9], this system of equations is sparse when the rating matrix is sparse, and this is the case in practice because each user will only rate a small number of items compared to the total number of items available.

As mentioned in [9], the minimum degree algorithm has a runtime of $O(n^3)$, where n is the number of users and items. In the typical case, many users have rated only very few items, and many items were only rated by few users. Because the corresponding nodes have small degrees, they are eliminated first, and in this phase of the algorithm, the runtime is linear.

7 Evaluation

We use two corpora for evaluation: MovieLens¹ and Jester². Each evaluation test consists of the following steps: First we choose a rating at random, then we remove this rating from the corpus. Afterward, we use a prediction algorithm to predict that rating. These steps are repeated for each of the two corpora, and for each of the following prediction algorithms:

Mean (M). The mean rating of the user

Pearson (P). The mean rating of other users weighted by their Pearson correlation to the user in question

Unit (U). The inverse resistance distance in the rating graph of unit resistances

Deep (D). The modified inverse resistance distance in the weighted rating graph

For each prediction method, we use linear regression (LR) to find an optimal affine function predicting the actual rating. We also use multiple linear regression (MLR) to predict ratings using different combinations of methods. For each combination, we calculate the mean squared error (MSE) and root mean squared error (RMSE)¹⁰. MSE and RMSE are *mean absolute error metrics* and the standard metrics to evaluate the predictive accuracy for recommender systems. For two vectors $x = (x_1, \dots, x_n)^T$ and $y = (y_1, \dots, y_n)^T$, the formula for MSE and RMSE is

$$RMSE(x, y) = \sqrt{MSE(x, y)} = \sqrt{\frac{\sum_{i=1}^n (x_i - y_i)^2}{n}} \tag{4}$$

The test results are shown in Figure 10.

Corpus	MovieLens			Jester		
	MSE	RMSE	LR/MLR function	MSE	RMSE	LR/MLR function
M	0.265	0.514	0.0023 +0.9928M	0.240	0.490	0.01 +0.799M
P	0.315	0.560	0.276+0.0005P	0.280	0.537	0.0325+0.059P
U	0.307	0.554	0.1299+0.0027U	0.292	0.541	-0.149 +0.0073U
D	0.285	0.534	0.236+3.03D	0.261	0.510	0.044+1.38D
M+D	0.239	0.489	-0.024+0.952M+2.81D	0.204	0.452	0.024+0.828M+1.478D
M+P+D	0.239	0.489	-0.024+0.952M+0.004P+2.84D	0.203	0.451	0.024+0.828M+0.022P+1.417D

Fig. 10. Evaluation results. Numbers indicate the MSE (mean squared error) and the RMSE (root mean squared error).

We observe that the Pearson correlation has a much smaller predictive accuracy than the other predictions, and that the mean rating is the best single prediction method tested. As users tend to give rating only within a certain range, for example only rating movies they like using exclusively positive ratings.

¹ <http://movielens.umn.edu/>

² <http://www.ieor.berkeley.edu/goldberg/jester-data/>

The mean rating and deep similarity combined perform better than the mean rating alone, and adding the Pearson based prediction does not lower the error.

8 Conclusion and Future Work

In the rating graph given by a rating system, we have defined a measure of similarity between nodes. This measure is based on work from [1] extended by the support for negative ratings.

The modified inverse resistance distance was shown to predict ratings more accurately than a Pearson correlation based algorithm, and using it in combination with other basic prediction methods gives better results than any method for itself.

The following areas of research remain to be explored.

- Formulate a modified Y- Δ elimination such that the modified resistance distance is preserved. As we have seen, the trivial algorithm does not give the desired results. This line of research would allow an implementation to work on a graph-based representation of the problem.
- Compare and combine the modified inverse resistance distance with other prediction methods. Many other collaborative filtering algorithms exist [3] and could be used in combination with our approach. Since we have seen that a combination of predictions can lead to a better prediction, this line may be promising.
- Analyze the complexity of calculating the modified inverse resistance distance and use methods such as clustering to reduce the rating graph size. As done with other prediction algorithms, the rating graph may first be reduced using methods such as clustering to improve the runtime.
- The modified resistance distance can be calculated between any two nodes in the graph. Calculating it between two users or items leads to a similarity (or distance) measure. This would be useful in recommendation systems and in social software when groups of similar users are to be detected.

References

1. Fouss, F., Pirotte, A., Saeens, M.: The application of new concepts of dissimilarities between nodes of a graph to collaborative filtering. *ACM Trans. Math. Softw.* (2003), Also TR-03-010 at www.cise.ufl.edu/tech-reports
2. Billsus, D., Pazzani, M.J.: Learning collaborative information filters. In: *Proc. 15th International Conf. on Machine Learning*, pp. 46–54. Morgan Kaufmann, San Francisco (1998)
3. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: *Proc. of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 43–52 (1998)
4. Mirza, B.J., Keller, B.J., Ramakrishnan, N.: Evaluating recommendation algorithms by graph analysis. *CoRR cs.IR/0104009* (2001)
5. Keller, B.J., Kim, S., Vemuri, N.S., Ramakrishnan, N., Perugini, S.: *The good, bad and the indifferent: Explorations in recommender system health*, San Diego, California (2005), <http://www.grouplens.org/beyond2005/full/keller.pdf>

6. Klein, D.J.: Resistance-distance sum rules. *Croatica Chemica Acta* 75(2), 633–649 (2002)
7. Qin, Z., Cheng, C.K.: Linear network reduction via Y- Δ -Transformation. technical report 2002-0706, University of California, San Diego, United States (2002)
8. George, A., Liu, W.H.: The evolution of the minimum degree ordering algorithm. *SIAM Rev.* 31(1), 1–19 (1989)
9. Heggenes, P., Eisenstat, S., Kurfert, G., Pothen, A.: The computational complexity of the minimum degree algorithm (2001)
10. 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)

Visual Query and Exploration System for Temporal Relational Database

Shaul Ben Michael and Ronen Feldman

Bar Ilan University, Department of Computer Science
Ramat Gan, Israel
{Shaybm1, Ronenf}@gmail.com

Abstract. This research is focused on developing effective visualization tools for query construction and advanced exploration of temporal relational databases. Temporal databases enable the retrieval of each of the states observed in the past and even planned future states. Several query languages for relational databases have been introduced, but only a few of them deal with temporal databases. Moreover, most users are not highly skilled in query formulation and hence are not able to define complex queries. The visual approach introduced here aims at simplifying the query construction process. It gives the user the option to define complex temporal constructs and provides visual tools with which to explore the returned networks intuitively. The exploration process should provide better insight into networks of entities, reveal patterns between the entities, and enable the user to forecast the behavior of entities in the future. A visual query language as an isolated subsystem is not sufficient in itself for a complete data analysis process. A query's output should be further explored to find patterns that are hidden in the output.

Keywords: visual query, temporal database, relational database, link analysis, text mining, data mining, social networks, risk management, data exploration, graph matching.

1 Introduction

The September 11 attacks brought into focus the malfunctioning of information agencies in the USA. In fact, many terrorist attacks could have been prevented if the available intelligence had been properly utilized. All the raw intelligence materials were available before these attacks were committed; the main problem was to push aside all the material that was not relevant to the investigation, to integrate all the relevant facts, and to construct a clear and complete picture of all the decentralized information available. In order to face these threats, intelligence agencies should adopt new strategies of data collection and data analysis. Text Mining is a new and exciting research area that tries to solve the information overload problem by using techniques from data mining, machine learning, NLP, IR, and knowledge management. Text Mining involves the preprocessing of document collections, information extraction, the storage of the intermediate representations, and techniques to analyze these intermediate representations. Link Analysis is the graphic portrayal

of extracted/derived data, in a manner designed to facilitate the understanding of large amounts of data and particularly to allow analysts to develop possible relationships between entities that otherwise would remain hidden by the mass of data obtained. The purpose of this research is to create a visual environment in which the end user is able to query and explore a temporal relational database. This research brings into focus methods and visual facilities that aid the user to extract relations according to their temporal behavior and to explore the evolution of the relations over time. The major assumption is that the analysis of static networks does not yield satisfactory information. If the time dimension is taken into consideration much more interesting conclusions can be inferred. Tracking relations over time may assist the user to identify special temporal patterns, to classify similar patterns, and to predict the future behavior of a relation. The system developed during this research is called pureVision in line with its designated purpose: to bring into focus uncharted trails hidden by the mass of data.

2 Related Work

Only a few research studies have dealt with the development of visual queries for relational data. Most visual query languages primarily deal with the query construction aspects and devote less attention to the results exploration aspects. Moreover, most of them do not support time-oriented properties. Visual query languages are used in various fields and aim to replace the traditional IR systems. These IR systems are designated for the retrieval of individual words or phrases ignoring the possible relationships between pieces of information within the text. Visual queries, however, are designated to identify complex patterns in raw digital information.

Multimedia is a good example where the standard methods of data retrieval are not sufficient. Digitized representation of images and video is usually heterogeneous and their content has semantic meaning, and hence a query based on simple comparisons of text/numerical values is not satisfactory. The user should be able to define complex structures of media components and their semantic relations. Several studies [1], [2], [3] proposed visual query interfaces for multimedia database management systems.

Other visual queries languages [4] take into consideration the structure of documents and enable more precise queries. These languages are not designated to mine the raw text as our language is, but rather only to extend the standard IR system by providing the option to look for textual items regarding structures unique to those items.

Our research deals with part of a group of visual query languages that are designated for knowledge discovery in relational database. Some of these studies treat static relational data [5] while others [6, 7, 8], like this work, try to expand the idea and develop methods to manage temporal relational databases. [6] introduced a Visual Language for Querying Spatial-Temporal Databases. [7] introduced Intelligent Visualization and Exploration of Time-Oriented Clinical Data. [8] addresses the issue of visual query formulation for temporal databases. This research introduces a number of visual constructs that allow the user to build queries in a modular fashion; however

it presents only theoretical issues and less effort has been devoted to the construction of a real system. Moreover tools for further data exploration are not used.

3 pureVision Architecture

The pureVision architecture enables incremental exploration for temporal relational data. Figure 1 illustrates the flow among the different stages of exploration in pureVision. pureVision analyzes temporal relational data as received from the text mining process and stored in a temporal relational DB. The system is divided into two main parts. The first part is dedicated to query management and the second to visual exploration. The analysis process may commence in two different ways. The user may use the visual query editor to construct his/her query or alternatively explore the output of previous queries. When the user has completed the query construction, the visual query engine translates the visual query into textual query. TLAL -Textual Link Analysis Language [9] is used as the textual query language. The textual query is used to search for graph matching in the DB. However, since TLAL does not support temporal patterns, the temporal extension is implemented. This module receives TLAL output and filters TLAL graph matchings according to the temporal constraints specified in the visual query. The exploration engine receives these graph matchings and displays them as networks of entities and relationships. The exploration engine provides visual tools that assist users to carry out advanced exploration and to elicit knowledge hidden inside those networks. All of the components mentioned here will be described in detail in the next chapters.

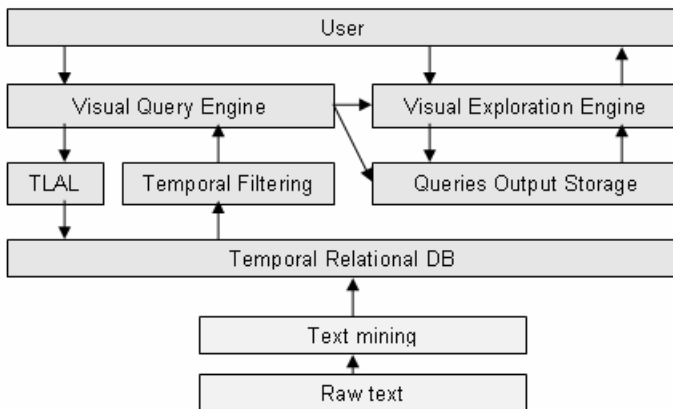


Fig. 1. Architecture of the pureVision system

4 Visual Query Engine

The visual query engine enables a visual construction of temporal queries. This kind of query enables users who are not highly skilled to define complex queries. The

language is capable of representing any type of graph structure, representing many kinds of temporal and non temporal constraints; additionally the language supports the definition of temporal patterns which aim to confine suspicious behavior of relations over time

4.1 Visual Query Components

The visual query engine includes several visual components which replace the components in the textual query. Figure 2 displays each of the visual components in the engine. The working window is specifically designed for the construction of visual queries; it functions as a space in which the user can drag visual elements and create the query graph. Ontologies define the hierarchical structure for a group of elements and they aid the user to navigate the query elements. The user can drag each element from a hierarchical structure directly into a working window and thus construct a query graph. The properties window enables the user to define constraints for relations and entities in the query graph. In a temporal database, each property is associated with a temporal field which represents the lifespan of the property. Therefore a temporal field is assigned to each property in the properties window. The temporal designer enables the construction of visual skits that represent a temporal pattern. Through the visual designer, the user may define time intervals in which a relation was active (the strength of a relation is positive) or passive (the strength of a relation is 0). Different rates of similarity can be applied to the temporal patterns constraint. The similarity rate determines the minimal correlation required between the skits and the temporal evolution of a relation as extracted from the database.

4.2 Temporal Patterns

Each relation has its own weight indicating the relation validity. Each relation is extracted from a raw text via the text mining process together with some statistical degree of confidence. The weight of each link reflects its validity and is stored in the DB. Tracing weight changes can reveal useful information about the nature of a relation. A weight of a relation is represented as a float number; in fact any non-zero value indicates that this relation is active (or feasibly active). The difference between non-zero values is less important compared to the difference between zero and non zero values; hence the problem is simplified by suggesting a binary weights presentation. Any non-zero value becomes 1 and all zero values remain the same.

Similarity tests for patterns. We use Pearson's correlation formula to measure the similarity between patterns with regular weights. We are interested in detecting linear dependencies between two temporal patterns where our underlying assumption is that the distribution of temporal patterns is normal (a temporal relation between entities mostly occurs in a specific period on a time scale where it tends to be rarer in times far from that period). Pearson's correlation formula is appropriate for this purpose. If the aim is to compare binary patterns, the number of occurrences when the two compared relations were active or passive at the same time can simply be counted, and the sum is divided by the total number of time points in the intersections of both patterns. High correlation between two patterns means that patterns are more likely to be similar.

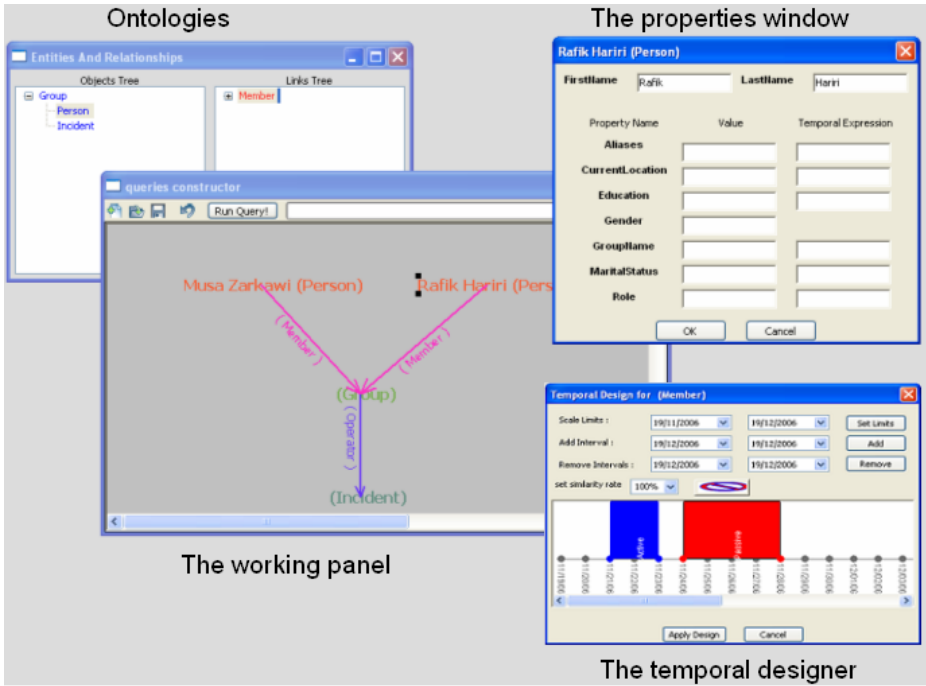


Fig. 2. The visual query components

5 TLAL API and TLAL Temporal Extension

5.1 TLAL API

The visual query engine presented in this thesis is built on the top of TLAL, an external library used in this system. The visual query is translated into TLAL scripts before any manipulation of the database can take effect. TLAL is designed to identify matches between patterns (such as the query graphs created by the visual query engine) and networks stored in a relational DB. TLAL is used rather than SQL because it is more suitable for handling textual data and for defining complex patterns (that are based on constrained networks)

5.2 Temporal extension for TLAL

The main and critical disadvantage of TLAL is its inability to manage temporal patterns. An extension for TLAL was therefore developed in order to cope with the deficiency. The aim of the temporal extension is to filter TLAL output according to the temporal patterns defined in the query. The process executed by the temporal extension is similar to the process TLAL carries out in order to find graphs in relational database. The target of this extension is to filter graphs that do not match the corresponding temporal patterns. Two methods are introduced here, developed for efficient graph pattern-matching.

PVNaive- pureVision semi naïve search method. PVNaive adopts the semi-naïve method for the graph pattern-matching problem. The algorithm guarantees that all of possible matches will be identified at the cost of higher time complexity. The algorithm starts with grouping together nodes according to their types, and then the algorithm filters the relevant target nodes for each node in the pattern according to the relevant constraints. PVNaive creates all permutations for the filtered target nodes in such a way that each permutation is a potential match, and then PVNaive filters the permutations that do not match the pattern links (all properties are checked except the temporal patterns). The final stage is devoted to filtering the matches that satisfy the temporal patterns. PVNaive is different from the basic naïve graph pattern-matching algorithm in that it does not generate every possible mapping from the n nodes in the pattern to the m nodes in the target, but rather tries to reduce the problem of the graph pattern-matching by generating a mapping from nodes in the pattern to the nodes in the target where both groups belong to the same type

PVBestFS - a based first search method. Best-first search is a search algorithm which optimizes breadth-first search by expanding the most promising node chosen according to some heuristic that attempts to predict how close we are to a solution. PVBestFS tries to expand nodes that lead to the exposure of as many graph matches as possible with minimal node expansion. The heuristic utilizes the number of edges that are connected to each node. The algorithm takes an initial node and tries to expand it in order to find complete graph matches. In order to choose the initial node the algorithm generates a mapping from nodes in the pattern to the nodes in the target where both groups belong to the same type and target nodes satisfy the pattern constraints. The most promising target group is chosen (according to the heuristic function). The algorithm ignores the direction of the links since the graph matching problem with undirected graphs is simpler. PVBestFS expands each of the nodes in the chosen group as explained in the following pseudo code. The algorithm works in an iterative way in order to identify complete matches for the structure of the pattern.

The pseudo code for the based first search method

```
PVBestFS (v)
1. Q <- insert (v, score=0, ancestors=nil)
2. While Q! =  $\emptyset$  do:
  a. Target Nodes <- pick the best group (Q)
  b. if Target Nodes cannot be expanded using the
  pattern do
    i. add ancestors to final solution
  c. Get links and descendant nodes (Target Nodes)
  d. create all permutations of Links and Descendants
  Nodes against the pattern
  e. for every permutation do:
    i. for all unexplored pNode and pLink in
```

```

permutation do:
  1. If pNode or pLink do not match the pattern
     invalidate the permutation
  ii. Score <- calculate (permutation). Assign the
     successor nodes a score using the evaluation
     function
  iii. If permutation matches the pattern: add to
     solution
  iv. Q <- insert (Descendants Nodes, score,
     ancestors)
f. POP (Q)

```

6 Visual Exploration System

The visual exploration engine enables the user to explore the results of the visual query. The visual exploration engine receives textual input that represents temporal networks; it provides visual tools that assist users to carry out advanced exploration and to elicit knowledge hidden inside those networks. There are two main exploration modes that can be adapted. The first mode deals with static graphs that exist at a given point in time. The second mode takes the temporal aspects of the networks into consideration, and gives a deeper insight into the evolution of temporal networks over time.

6.1 Temporal Networks Formalization

The formalization introduced here is inspired by the TEER data model [8]. A temporal network can be represented by a temporal graph $G(E, L)$, where E is a set of entities and L is a set of temporal links between the entities.

Let T be an accountably infinite set of totally ordered discrete points in time. A time interval $[ts, te]$ is defined to be a set of consecutive points in time; that is, the totally ordered set:

$$\{ts, ts+1 \dots te\} \subset T \quad (1)$$

A temporal element, denoted as TE , is a finite union of time intervals, denoted by $\{I_1 \dots I_n\}$, where I_i is an interval in T .

A link type L of degree 2 has two participating entity types, E_1, E_2 . Each link instance l in L is a 2-tuple $l = \langle e_1, e_2 \rangle$ where each $e_i \in E_i$. Each link instance l is associated with a temporal element $TE(l)$ which gives the lifespan of the link instance. $TE(l)$ must be a subset of the intersection of the temporal elements of the entities e_1, e_2 that participate in l . Formally,

$$TE(l) \subseteq (TE(e_1) \cap TE(e_2)) \quad (2)$$

This is because, for the link instance to exist at some point t , all the entities participating in that link instance must also exist at t .

6.2 Temporal Networks Display and Exploration

The main exploration frame can display networks in two ways: a sliced network of relations defined in a specific time point and a complete network that includes the

union of relations in each time point. Formally A link $l = \langle e1, e2 \rangle$ exists at a time point t iff $t \in TE(l)$. In complete networks only one representative relation is drawn for each pair of entities. The user can navigate between temporal slices using the visual interface. A temporal network may be displayed in two modes: the normal mode and the 'difference' mode. Relations displayed in normal mode are colored by their type, and the color in that mode is not affected by the state of a relation in previous or following points in time. The 'difference' mode, however, aims at highlighting the changes of relations activities in different time slices. Relations that do not appear in the preceding display are painted in a different manner.

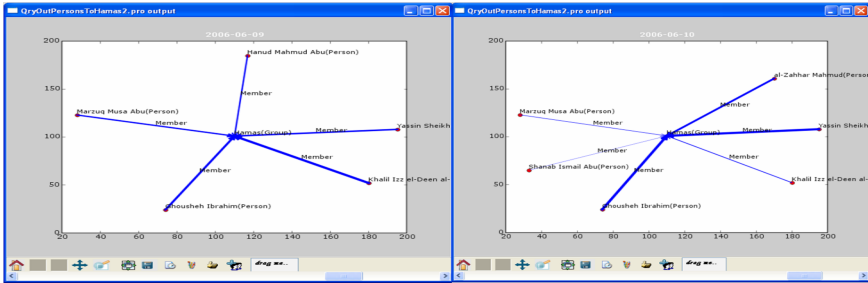


Fig. 3. Navigation in temporal network

During the exploration process the user may notice a suspicious entity or relation and may be interested in investigating it. Clicking on any element in a network exhibits a list of properties with their values. Additionally, several optional operations are introduced to the user. The user may investigate an entity in two ways: find the shortest path from the source entity to another target entity specified by the user, and find all paths according to a specified depth. Shorter paths between entities may reveal the significant semantic relations between them. The user may investigate the evolution of a relation over time. A special sub-frame is dedicated to the display of the degree of confidence for each time point when the relation was observed. The exploration of the relation's evolution may expose suspicious patterns. Circular patterns, for example, can highlight the periods when the relation is active and aid the user to predict when the relation is going to be active in the future.

The quality of temporal networks is determined in particular by their reliability. Relations characterized by a lesser degree of confidence are not as interesting as relations with a higher degree of confidence. There are two ways in which the user can easily spot the difference in relations reliability. First, temporal networks are visualized in such a way that more reliable relations are distinguished by broader links. Second, the relations can be ranked by their reliability and shown in a different frame as a ranked list.

The size of explored networks tends to be correlated with the size of the relational database and with the generality of the queries. Special features are introduced in order to manage large networks. The user may focus on a sub-network of special interest by zooming into it or alternatively zooming out to see a more general view of relationships. The pan feature enables the user to move networks from their original position to another position where they do not fit the size of the panel in which they

are displayed. Every manipulation carried out on the visual networks is saved, and hence the user can retrieve previous states of the network display.

Multiple working frames. A working frame is supposed to handle a temporal network as returned from a single query. However, it is frequently necessary to compare the output of different queries. Here are introduced the intersection and union methods for temporal networks. Let us define source graphs G_1 , G_2 and result graph G_{res} as a pair (E, L) where E is a set of entities, and L is a set of temporal links between those entities. G_{res} is the intersection of G_1 and G_2 if it satisfies the following conditions:

$$\text{Link } l = \langle e_1, e_2 \rangle \in L \leftrightarrow l_1 = \langle e_1, e_2 \rangle \text{ and } l_2 = \langle e_1, e_2 \rangle \text{ and } TE(l_1) \cap TE(l_2) \neq \emptyset \text{ and Type}(l_1) = \text{Type}(l_2) \quad \forall l_1 \in L_1 \quad \forall l_2 \in L_2 \quad (3)$$

$$\text{Entity } e \in E \leftrightarrow \langle e, x \rangle \in L \text{ or } \langle x, e \rangle \in L \quad \forall x \in E. \quad (4)$$

The union graph G_{uni} can be defined as a pair (E, L) where

$$L = L_1 \cup L_2 \text{ and } E = E_1 \cup E_2 \quad (5)$$

$$l_1 = \langle e_1, e_2 \rangle \text{ and } l_2 = \langle e_1, e_2 \rangle \text{ and } \text{Type}(l_1) = \text{Type}(l_2) \rightarrow TE(l_1) \cup TE(l_2) \quad \forall l_1 \in L_1 \quad \forall l_2 \in L_2 \quad (6)$$

The user can carry out these operations by dragging one working frame on another and choose the appropriate operation.

6.3 Temporal Clustering

In general, a cluster is defined as a set of similar objects. This "similarity" in a given set may vary according to data. We focus here on the temporal clustering problem, our purpose being to group together relations with similar temporal behavior. The identification of relations with a high similarity rate can bring into focus hidden connections between entities that are not explicitly connected within the network structure. The distance between two relations is determined by the correlation function mentioned above. There is an opposite ratio between the distance function and the correlation function, since highly correlated relations should belong to the same cluster.

DBSCAN - density based spatial clustering of applications with noise [10] is used as the clustering algorithm. DBSCAN relies on a density-based notion of clusters which is designed to discover clusters of arbitrary shape. DBSCAN performs good efficiency on large databases. DBSCAN requires only two input parameters, density reachability and density connectivity. These concepts depend on two parameters: ϵ - the epsilon (radius) neighborhood of a point, and minp - the minimum number of points in the epsilon neighborhood. Density reachability is defined as follows. A point p is density reachable from point q if the two following conditions are satisfied: first the points are close enough to each other - $\text{distance}(p; q) < \epsilon$ and second there are enough points in q neighborhood. Density connectivity is defined as follows. A point p is density-connected to a point q if there is a point o such that both, p and q , are density reachable from o .

Clusters are not displayed on a static panel. Like the networks display, the clusters display can be manipulated in order enable the user to explore the clusters. The user can click on any relation and explore its evolution over time; this facility directs the user towards comparing the evolution of different relations from the same cluster as well as from different clusters.

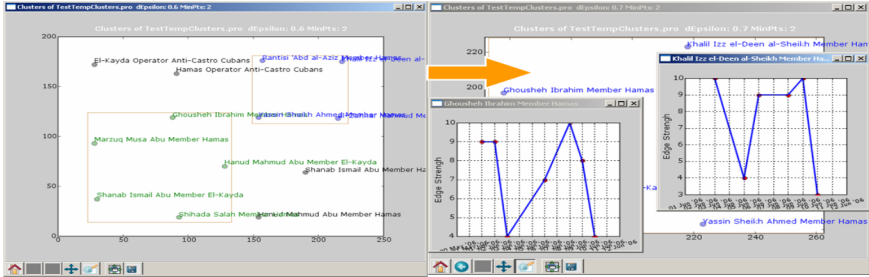


Fig. 4. Illustration of zooming into a cluster and the exploration of the clusters relations

6.4 The Graph Drawing Problem

There are many ways to visualize the structure of networks. The optimal visualization is the clearest to the user. The graph drawing problem is simpler in our domain because we treat here drawings in which edges are drawn as straight lines and the purpose is to determine the position of the vertices. The implementation of Kamada and Kawai [11] is used in order to solve the graph drawing problem. Kamada and Kawai's is a simple but successful algorithm for drawing undirected weighted graphs. The basic idea of Kamada and Kawai is as follows. The desirable “geometric” (Euclidian) distance between two vertices in the drawing graph is the “graph theoretic” distance between them in the corresponding graph. The algorithm introduces a virtual dynamic system in which every two vertices are connected by a “spring” of such a desirable length. Then the optimal layout of vertices is regarded as the state in which the total spring energy of the system is minimal.

7 Evaluation

We are interested in examining the performance of PVNaive and PVBESTFS on different kind of inputs. We are primarily interested in examining how these algorithms compare on complex query graphs, and how they compare on different kinds of networks that are processed by them. We designed two experiments to test each of these goals. For the first experiment several queries were constructed which varied in their graph complexity where query1 is the simplest and query5 is the most complex. The more edges in the query graph, the more complex it is considered to be.

pureVision receives temporal networks that are returned by TLAL and uses PVBESTFS and PVNaive in order to filter those networks according to the temporal constraints specified in the query. However, we want to isolate the impact of the query graph complexity so that the variable size of the networks will not impact the results. Therefore we used the same simulated output for all queries scripts (instead of the output of TLAL)

and no constraints were set in the queries because constraints may prune the search space of the algorithms and we want to isolate the impact of query graph complexity.

Figure 5 shows that PVBESTFS achieved better results for all observations. The gap between the results is more remarkable for more complex patterns. There is a correlation between the patterns complexity and the ratio between the time performance of PVBESTFS and PVNAIVE. PVBESTFS can cope with complex graphs structures more efficiently, and this advantage is important when the user is interested in constructing complex queries and the networks returned from TLAL are too large to be managed by the naïve method.

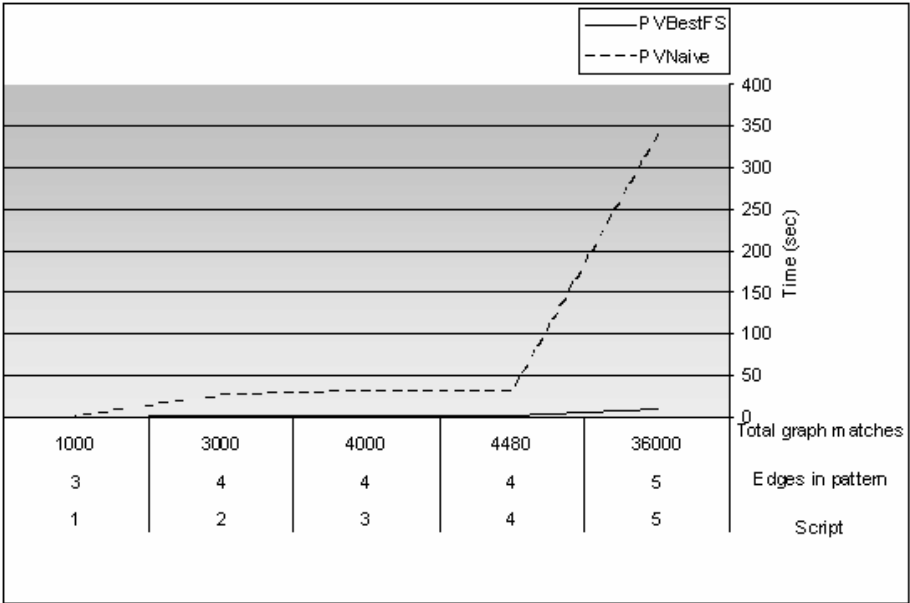


Fig. 5. The Time complexity of PVBESTFS and PVNAIVE as a function of query graph complexity

For the second experiment two sets of networks were constructed. The first set of networks is designated to test the impact of temporal links on performance. In this set all of the networks have the same nodes, and the networks are varied in the number of temporal links between each two nodes (each link represents different time). The same query graph is used in all the tests since we want to isolate the impact of temporal factor of the tested networks.

Figure 6 shows that the time complexity is almost linear for both of the algorithms and the form of the graphs is almost identical. The meaning of this finding is that temporal links have no significant impact on the difference between the performances of PVBESTFS and PVNAIVE.

The second set of networks is designated to test the impact of network density on performance. Network density measures the ratio between quantity of links in a given network and a full network (as received from the given network).

It can be seen in Figure 7 that network density has a great impact on the difference between the performances of the algorithms. PVBESTFS achieved much better results when tested on low density networks where the results of PVNaive showed minor changes.

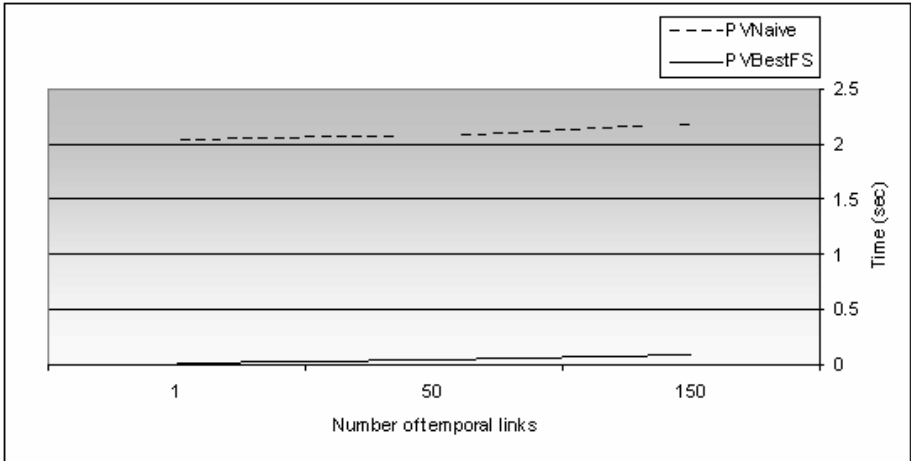


Fig. 6. Time performance of PVBESTFS and PVNaive as a function of the temporal links number

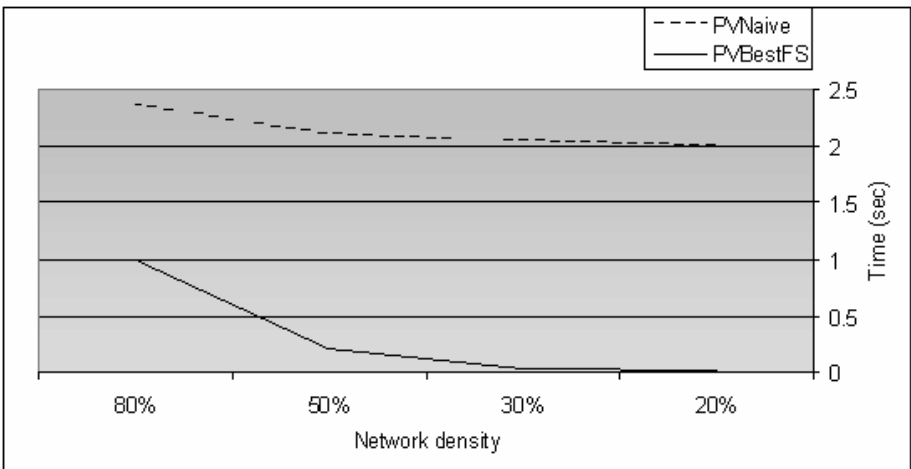


Fig. 7. Time performance of PVBESTFS and PVNaive as a function of network density

The second experiment shows that PVBESTFS can handle networks with medium or lower density much more efficiently than can PVNaive. In practice these kinds of networks are much more frequent and thus this finding is notable; however PVBESTFS had no remarkable advantage over PVNaive when the temporal factor

was tested. Only the non temporal structure of networks (where temporal links are not taken into consideration and each pair of nodes can be connected by a single link or not) results in an increasing gap between the performances.

8 Discussion

In this research study the issue of visual query language and advanced exploration systems for temporal relational data is addressed. One complete system was developed that enables the user to construct complex queries and to identify suspicious patterns in temporal relational data in an incremental fashion. Using pureVision, even naïve users are able to discover suspicious patterns in a significant pool of data. This research is part of the link analysis field and it aims at developing disciplines to explore relational data that are returned from the text mining process. The research was focused on the temporal aspects of relations. pureVision makes it possible to search for relations according to certain temporal patterns defined by the user, to investigate the evolution of relations over the time, to identify suspicious behavior of relations, to cluster relations with similar behavior, and to approximate the behavior of relations in the future. Advanced graph matching algorithm was developed to improve the time performance for complex queries and for medium and low density networks. These algorithms are designated to make pureVision more interactive system.

References

1. Yoshitaka, A., Ichikawa, T.: A Survey on Content-Based Retrieval for Multimedia Databases. *IEEE knowledge and data eng.*, pp. 81–93. IEEE Computer Society Press, Los Alamitos (1999)
2. El-Medani, G.: A Visual Query Facility for Multimedia Databases, University of Alberta, Technical Report, pp. 18–95 (1995)
3. Oria, V., Xu, B., Cheng, L.I., Iglinski, P.J.: VisualMOQL: the DISIMA Visual Query Language, *Multimedia Computing and Systems*, pp. 536–542 (1995)
4. Baeza-Yates, R., Navarro, G., Vegas, G., De La Fuente, P.: A model and visual query language for structured text, *String Processing and Information Retrieval*, pp. 7–13 (1998)
5. Blau, H., Immerman, N., Jensen, D.: A visual query language for relational knowledge discovery, University of Massachusetts Amherst, Technical Report, pp. 1–28 (2001)
6. Bonhomme, C., Trepied, C., Aude-Aufaure, M., Laurini, R.: A Visual Language for Querying Spatio-Temporal databases, pp. 34–39. ACM Press, New York (1999)
7. Shahar, Y., Cheng, C.: Intelligent Visualization and Exploration of Time-Oriented Clinical Data. *The 32nd Annual Hawaii International Conference*, pp. 15–31 (1999)
8. Kouramajian, V., Gertz, M.: A visual query language for temporal databases, Hannover, pp. 388–399 (1995)
9. Feldman, R., Ozz, R.: Link Analysis in Networks of Entities, Technical Report, Bar-Ilan University (2007)
10. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *KDD'96*, pp. 226–231 (1996)
11. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs, pp. 7–15. Elsevier, Amsterdam (1989)

Towards an Online Image-Based Tree Taxonomy

Paul M. de Zeeuw, Elena Ranguelova, and Eric J. Pauwels

CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
paul.de.zeeuw,elena.ranguelova,eric.pauwels@cwi.nl

Abstract. This paper reports on a first implementation of a webservice that supports image-based queries within the domain of tree taxonomy. As such, it serves as an example relevant to many other possible applications within the field of biodiversity and photo-identification. Without any human intervention matching results are produced through a chain of computer vision and image processing techniques, including segmentation and automatic shape matching. A selection of shape features is described and the architecture of the webservice is explained. Classification techniques are presented and preliminary results shown with respect to the success rate. Necessary future enhancements are discussed. Benefits are highlighted that could result from redesigning image-based expert systems as web services, open to the public at large.

1 Introduction

The pervasiveness of broadband Internet and mobile telephony has created an unprecedented connectivity between people and computational devices such as computers, mobile phones, digital camera's and GPS units. As a consequence, a growing number of initiatives is harnessing this infrastructure to set up new communities and exploit novel opportunities for large-scale interaction and participation. As Internet access thresholds continue to fall, the public at large is slowly being transformed from passive content consumers into active and avid content producers. Indeed, the likes of Wikipedia, Flickr, and YouTube have demonstrated beyond any doubt the viability of "crowd sourcing" development projects (a.k.a *Peer Productions*, e.g. Yahoo Answers) in which a comprehensive, high quality product emerges as the result of modest contributions from literally thousands or even millions of participants. This goes to show that there is a tremendous amount of talent and resources "out there" of people who both have the means and the aspiration to contribute to online communities that have captured their interest.

In this paper we report on our ongoing efforts to set up a (to the best of our knowledge, first!) *web-based tree taxonomy searchable by image query*. More specifically, we have created a webservice¹ that aspires to assist users in identifying trees by uploading a photograph of one of their leaves. This photograph is then processed by the web server and matched against a database of exemplars

¹ http://biogrid.project.cwi.nl/projects/leaves_v2/

of known species. As a result, a web page is created showing the, say, ten most similar exemplars along with species information, inviting the user to make the final choice and provide feedback. If the user considers the determination to be successful, the image is retained as a new exemplar for future queries. If the user deems the identification to be unsuccessful, the image is forwarded to an expert-biologist for a second opinion.

We envisage that in the near future it will become possible to point the camera in your mobile phone at a plant or tree, take a snapshot of one of the leaves and send it as an MMS message to a designated phone number, such as 1-800-whichtree, say. Half a minute later you receive an sms serving up both the Latin and common name for the tree, as well as the link to the Wikipedia page where more information can be found. Moreover, it is our explicit intention to open this webservice up to the public at large so that *in addition to querying* everyone can *contributeo* to the exemplar database by uploading information and images of tree species.

1.1 Previous and Related Work

Although there are a number of online tree taxonomies available, the proposed web service is, in our opinion, innovative to the extent that it supports image-based queries and therefore contributes to the small but growing collection of applications that try to extend Internet search beyond the classic keyword paradigm. The best-known web-applications that support similar input modalities are focusing primarily on face recognition, such as FaceIt, myheritage.com or Riya.

It is clearly apparent that, given the pervasiveness of digital cameras, biologists are waking up to the possibilities of computer-assisted photo-identification, and a number of stand-alone systems are under development (cf. [13,6,7]). However, to the best of our knowledge the proposed website is the first to offer a vision-based taxonomy system as a web service.

2 Architecture of the Webservice

Broadly speaking the webservice is designed as a 2-tier system: The front-end allows the user to upload query images which are then shipped to the back-end server for processing and matching. The results are included in a webpage which is then transferred to the front-end for display and feedback. The system therefore comprises the following main components:

- **Database of exemplars on back-end.** Predictably, one starts by creating a database of *exemplars* that encapsulate the domain knowledge — for the taxonomy application we will often refer to them as *exemplars*. This database contains, for each of the target tree species, information such as the common name (e.g. *white oak*), the genus and species as specified in the Linnaeus binomial nomenclature (e.g. *Quercus alba*), a link to Wikipedia (if available), as well as one or more relevant photographs. Associated with

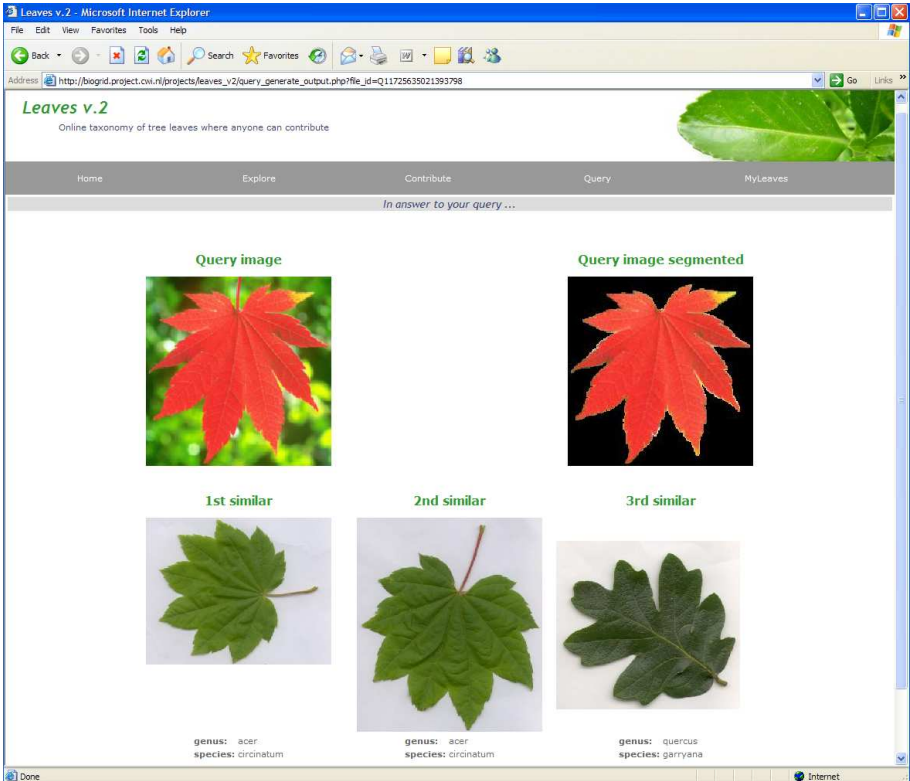


Fig. 1. Webpage generated by the taxonomy webservice in response to a submitted query image (top left). The result of the automatic segmentation is shown top right. The most similar images in the cases-database are displayed on the second row, together with relevant metadata such as genus and species. The displayed shortlist of most similar leaves offers the user the possibility to pick – as a final selection – the leaf that best matches his query image.

each photograph is a set of automatically computed numerical features that characterize the shape of the corresponding leaf (for more details on these features, see section 3).

- **Front-end.** The front-end is a straightforward webpage that allows the user to upload an image of the query leaf. To improve performance we request users to adhere to certain standards (e.g. the leaf should be photographed against high contrast background) which greatly increase the reliability of the automatic image segmentation.
- **Processing on back-end.** Uploading a query image triggers a sequence of algorithms that (i) segment the leaf from the background and extract the result as a binary mask, and (ii) computes ten numerical shape features (for more details, see Section 3). The results are then checked against the pre-computed exemplar features and the most similar ones are shortlisted.

- **Feedback** . Once the similarity-based shortlist is available, a response webpage is compiled displaying images of the n best matches (where n typically ranges between 3 and 10). Each image has a caption detailing the genus, species and common name of the corresponding tree. Showing a ranked shortlist of images (see Fig. 1) allows the user to perform a final visual check and discard any obvious mismatches.

3 Features for Shape Matching to Exemplars

3.1 Image Segmentation: Segregating Foreground from Background

Prior to computing the features detailed below, both exemplar and query images are first segmented into actual leaf (foreground) and a background. In what follows, we will use the term *mask* to refer to the resulting binary image that specifies the foreground pixels. Note that we can think of leaves as flat objects (something everyone who assembled a herbarium book can relate to) which simplifies the analysis of the image considerably as we can restrict our attention to measures for 2D shapes.

Automatic leaf segmentation proceeds through a number of steps. First, the colour image is converted to gray scale in several different manners (using the RGB and the HSV values). Then, each gray-level representation is segmented using a gray-level histogram to which a mixture of Gaussian density is fitted. The local minimum of the density is then used as a data-driven threshold for segmentation. In this manner, several initial segmentations are obtained. For each binary segmentation the number of 1 - connected components is computed and the best initial segmentation is chosen as the one with minimum number.

The next step of the algorithm uses the initial segmentation to guide a *watershed transformation* on the best gray-level representation of the original image. The watershed transformation is a powerful and well-established mathematical morphology tool for image segmentation which has been used in many applications [5]. Any grey-level image can be considered as a topographical surface. Flooding this surface from its minimum while preventing the merging of water coming from different sources, will result in a partitioning of the image into catchment basins associated with each minimum. The boundaries between the catchment basins are the watershed lines. If we apply this transformation to the gradient of an image, we should obtain catchment basins corresponding to homogeneous grey-level regions. It is well-known however, that the transform tends to produce an over-segmentation due to the local variations in the gradient. A *marker-controlled transformation* is a solution to this problem: The gradient image is modified via morphological reconstruction [5] in order to keep only the most significant gradient edges in the areas of interest between the markers. The biggest connected component from the chosen initial segmentation is used as the *foreground marker* and the image boundaries as the *background marker*. As a result of this step one gets the leaf boundaries.

The last step is the stem detection and removal. The stem is considered as a significant deviation from the main leaf shape. All such deviations are detected

using the *top-hat transform* [5] of the binary segmentation. The detected deviations are the "teeth" of the leaf margin and the stem. The stem is singled out by imposing the additional restriction for large eccentricity and major axis. After the stem removal, the remainder is used as the binary leaf shape mask.

3.2 Shape Matching

Shape – even planar shape – is notoriously tricky to characterize accurately in a short sequence of numerical features. In order to cope with the considerable variations encountered in the dataset, we have implemented a number of features, each tailored to capture specific shape aspects. The idea is that combining them will produce a more discerning similarity measure. Below we briefly discuss the selection of features that are currently being used. It is likely that this set will be expanded in future versions of the search engine. All of them are computed on the binary image (a.k.a. mask) that results from the segmentation. This means that all internal structure, such as colour, texture and (most importantly) vein-structure has been discarded. This is an obvious weakness in the current system that we intend to remedy in a subsequent version. We also assume that the stem has been pruned so that only the intrinsic leaf shape remains. For ease of future reference we denote by L the resulting 2-dimensional shape, and by ∂L its contour.

Solidity (Sol). To measure the extent to which a leaf is lobed, we compute its *solidity* which is defined by comparing the area of the leaf to the area of its convex hull ($CH(L)$) to obtain a number between 0 and 1:

$$\text{Sol} = \frac{\text{area}(L)}{\text{area}(CH(L))}.$$

Isoperimetric factor (IF). This is another measure that roughly captures how winding (oscillatory) the contour is. If the perimeter is defined as $\ell = \text{length}(\partial L)$ and $A = \text{area}(L)$ then IF is defined as

$$IF = \frac{4\pi A}{\ell^2} \leq 1.$$

Equality prevails if and only if the contour is a circle.

Eccentricity (X). The third straightforward measure we employ is the eccentricity of ellipse with identical second moment as the leaf shape L .

Moment Invariants for Shape Characterization. The next set of measures are less straightforward. Hu's invariants [2], based on centralized moments, serve as a classic tool for recognizing geometrical shapes. An image is regarded as a density distribution function f . A central moment $\mu_{pq}(f)$ of f is given by

$$\mu_{pq}(f) = \iint_{\mathbb{R}^2} (x - x_c)^p (y - y_c)^q f(x, y) dx dy, \quad (1)$$

where p and q are non-negative integers and (x_c, y_c) is the center of mass. Such a central moment is said to be of $(p+q)$ th order. In our case a binary foreground mask plays the role of the image of f which equals 1 inside the leaf region and 0 outside of it. By their definition it is immediate that central moments are translation invariant. Hu constructed polynomials with variables μ_{pq} in such a way that the outcome is invariant under rotations and reflections (the latter apart from sign). Two polynomials are built with second-order moments, four polynomials with third-order moments and one combines second-order and third-order moments.

$$I_1 = \mu_{20} + \mu_{02}, \quad (2)$$

$$I_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2, \quad (3)$$

$$I_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2, \quad (4)$$

$$I_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2, \quad (5)$$

$$I_5 = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2), \quad (6)$$

$$I_6 = (\mu_{20} - \mu_{02})((\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}), \quad (7)$$

$$I_7 = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2). \quad (8)$$

We elaborate briefly on the numerical computation of the moments. Using the values of the image pixels we construct an interpolating function based on piecewise constant approximation. The piecewise constant basisfunctions have their support on squares centering around the pixels. Furthermore, the rectangular domain of an image is scaled in the sense that the size of the short side is equal to 1. As the supports of the basisfunctions are square, the size of the longer side of the domain follows at once. Hereby we can now perform the integration in (II) numerically.

So far, the expressions defined by (2)–(8) using (I) are invariant under translation, rotation and reflection (provided we ignore the sign of I_7). For shape invariance we still need to enforce similitude invariance, that is, after a mere change in dimensions of an object (leaf) it is identified as the same. Such invariance can be obtained by normalizing the moments μ_{pq} . Dilations (changes in size) by a scalar $\alpha > 0$ of the whole image or of objects in an image against a neutral background result in new central moments given by [2]

$$\mu'_{pq} = \alpha^{p+q+2} \mu_{pq}. \quad (9)$$

It follows in particular that $\mu'_{00} = \alpha^2 \mu_{00}$, and also $\mu'_{20} + \mu'_{02} = \alpha^4(\mu_{20} + \mu_{02})$. Combining this result with Eq. (9) yields

$$\frac{\mu'_{pq}}{(\mu'_{00})^{(p+q+2)/2}} \frac{\mu_{pq}}{\mu_{00}^{(p+q+2)/2}}, \quad \frac{\mu'_{pq}}{(\mu'_{20} + \mu'_{02})^{(p+q+2)/4}} \frac{\mu_{pq}}{(\mu_{20} + \mu_{02})^{(p+q+2)/4}}$$

respectively. As we recall that both μ_{00} and $\mu_{20} + \mu_{02}$ are invariants with respect to rotation and reflection this demonstrates how to normalize the moments to achieve invariance under dilation. The first choice leads to the following new set of invariant generators [4]

$$\begin{aligned} I'_1 &= I_1/\mu_{00}^2, & I'_2 &= I_2/\mu_{00}^4, & I'_3 &= I_3/\mu_{00}^5, & I'_4 &= I_4/\mu_{00}^5, \\ I'_5 &= I_5/\mu_{00}^{10}, & I'_6 &= I_6/\mu_{00}^{10}, & I'_7 &= I_7/\mu_{00}^7. \end{aligned} \tag{10}$$

The second choice leads to a different but similar result. It may be more suitable (as a starting point) in case the density distribution corresponds to wavelet detail coefficients, see [4].

Finally, it is clear that the *shape* of the foreground in the binary image f should be invariant under a change in luminosity; in mathematical parlance: $f \mapsto \lambda f$ where $\lambda > 0$. As a scalar multiplication of the distribution function f does not affect the center of mass, it follows from [10] that

$$\mu_{pq}(\lambda f) = \lambda \mu_{pq}(f), \text{ for all } \lambda \neq 0. \tag{11}$$

One observes that the feature vector I' defined element by element through [10]

$$(I'_1, I'_2, I'_3, I'_4, I'_5, I'_6, I'_7)$$

then changes into

$$(\lambda^{-1}I'_1, \lambda^{-2}I'_2, \lambda^{-3}I'_3, \lambda^{-3}I'_4, \lambda^{-6}I'_5, \lambda^{-4}I'_6, \lambda^{-6}I'_7)$$

which is an undesirable result. (The result when moments would not be normalized would be equally undesirable.) To overcome this inhomogeneous change in the feature vector we use the following operator

$$R_p(u) = \text{sign}(u)|u|^{1/p}, \text{ with } p \in \mathbb{N} \text{ and } u \in \mathbb{R}. \tag{12}$$

When applied to an invariant I_k it produces again an invariant. It is a "legal" operation that invariants can be subjected to, i.e., neither their invariance properties nor their discriminative power are lost. We define the homogenized feature vector as

$$\tilde{I}' = (I'_1, R_2(I'_2), R_3(I'_3), R_3(I'_4), R_6(I'_5), R_4(I'_6), R_6(I'_7)). \tag{13}$$

This feature vector \tilde{I}' now satisfies the *homogeneity condition* [4][8] in that a rescaling of the luminosity now affects all components in a homogeneous fashion:

$$f \mapsto \lambda f \implies \tilde{I}' \mapsto \lambda^{-1} \tilde{I}'. \tag{14}$$

In addition, it turns out that hereby all elements operate in the same order of magnitude and that Mahalanobis's method is superfluous.

4 Results and Discussion

4.1 Populating the Exemplar Database Through Webharvesting

For the above outlined system to be successful, it needs to have access to a sufficiently large and comprehensive database of exemplars (cases). In order to create this with minimal effort we have taken recourse to webharvesting. More precisely, using Wikipedia we have first compiled a long list of the (Latin) Linnaeus classification (i.e. *genus* and *species*) of the tree species we are interested in. This list is fed into a programme that submits each combination into Google's *Image Search* and collects all the images that are returned. Relatively straightforward image processing software then winnows down this collection by rejecting all pictures that lack a convincing oval-shaped foreground. In most cases this prunes the collection down to a few percent of the original "harvest". The final selection is done by a human supervisor who reject everything except images where the well-defined foreground corresponds to a leaf. These images then go in the exemplar database and the Linnaeus classification (i.e. the *genus* and *species* that were used as search terms) are inserted as metadata. Once these images have been added to the exemplar database, the residing feature agents jump into action and compute the necessary features so that these new exemplars can be compared to any incoming image queries.

As mentioned earlier we have realised a first implementation of the above outlined webservice. To date we have compiled a small database of exemplars which comprises 23 unique genus-species combinations harvested from the web. For each of these genus-species combinations we have on average 5 to 10 exemplar images, adding up to 146 images in total. All the images have been segmented and the above-mentioned 10 shape parameters have been computed (i.e. solidity, isoperimetric factor, eccentricity and 7 moment invariants). We have then tested the two classification tools which we describe next.

4.2 Classification Trees

Classification trees seemed a first obvious choice for the classification of leaves (no pun intended!). The full 10-dimensional feature vector of section (3.2) was used to predict class-membership (running from 1 through 23 as there are 23 unique genus-species combinations). However, when we tested performance using cross-validation, the prediction success turned out to be disappointingly low 42%. For that reason we switched to a nearest neighbour classifier described in the next section.

4.3 Nearest Neighbour Classification

Since the 10-dimensional feature-vector is an amalgamation of qualitatively different characteristics (dimension-wise $10 = 1+1+1+7$), we decided that it was best to first compute distances in each space separately, and then produce a resulting distance by computing an (empirically optimized) linear combination

of these partial results. We settled for the straightforward (1-dimensional) Euclidean distance for the solidity, eccentricity and isoperimetric features (denoted by d_{Sol} , d_X and d_{IF} respectively). Further, because of (14) we opted for the (normalized) cosine distance in the 7-dimensional space of homogenized moment invariants:

$$d_{HM}(\vec{x}, \vec{y}) = \frac{2}{\pi} \arccos \left(\frac{|\langle \vec{x}, \vec{y} \rangle|}{\|\vec{x}\| \cdot \|\vec{y}\|} \right).$$

All distances are within the 0–1 range, simplifying comparison and combination.

We proceed by making the assumption that the comprehensive distance is a straightforward linear combination of the above:

$$d_{LC} = d_{HM} + \alpha d_{Sol} + \beta d_X + \gamma d_{IF}.$$

The values for the weight parameters are determined by systematically searching for the combination that produces the best results on the exemplar database, i.e. each exemplar is used as a test-image and assigned to the same class as its nearest neighbour (in the d_{LC} -metric). The discriminative power of d_{HM} turns out to be predominant but even though $\alpha, \beta, \gamma \ll 1$ the other distances cannot be dispensed with.

We have estimated the classification accuracy by simulating the results that would be displayed on the webpage. More precisely, for each exemplar we have computed the 10 nearest d_{LC} -neighbours as these would be displayed on the webpage if the selected exemplar was submitted as a query. The results are shown in Fig. 2. If we insist that classification is only successful if the most

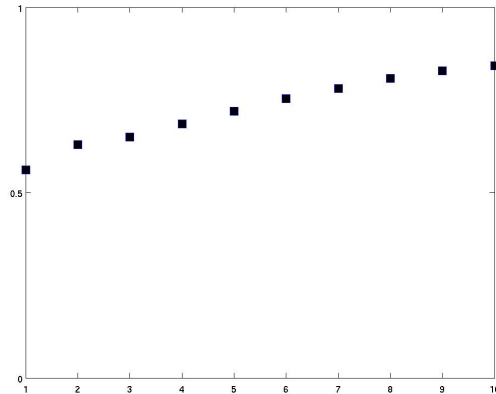


Fig. 2. Query success rate in terms of the number (k) of retrieved nearest neighbours. If we allow the user to inspect the ten most similar images, then the success rate is slightly higher than 85%. See main text for more details.

similar has the correct genus and species, then the success rate is about 53%. However, this is unduly pessimistic as not one but ten nearest neighbours are

shown on the webpage, and the user is invited to manually pick the best match. This means that the query will be successful if the correct genus and species are found among the 10 nearest neighbours (i.e. $k = 10$ in Fig. 2). It turns out that for this less stringent success-criterion, the success rate is approximately 85%. The complete distance matrix is shown in Fig. 3.

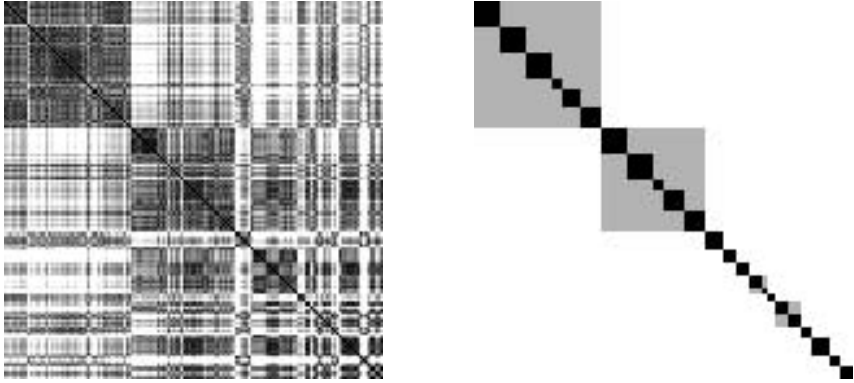


Fig. 3. *Left:* Schematic representation of the (146×146) d_{LC} -distance matrix for the database of exemplars. Bright points represent large distances while dark shades indicate similarity. *Right:* The corresponding class delineation: black blocks corresponds to leaves that have identical genus and species, while gray shades indicate identical genus but different species within that genus. Ideally, the d_{LC} matrix on the left should look very similar to the classification groundtruth depicted on the right.

5 Conclusions and Future Directions

In this paper we have reported on a first implementation of a webservice that supports image-based queries within the domain of tree taxonomy. We have argued that thanks to computer vision and image processing it is now feasible to gain good segmentation and recognition results with little or no input from a human supervisor. This opens the door to efficient searches through large databases of photographic material and therefore allows us to tackle classification problems for which the domain knowledge is primarily encoded in visual form. These developments are effectively extending the scope of image-based search task where traditionally, most efforts have been focused on face recognition. Clearly, photo-identification of plant and animal species (individual animals even) opens up a vast and exciting new application domain, the relevance of which is beyond dispute given the current concerns about the conservation and biodiversity.

The preliminary results that are obtained seem acceptable but clearly leave considerable scope for improvement. Apart from employing a much larger dataset, we see at least two directions that remain to be explored. First, we could add more sophisticated shapes measure to fine-tune the global distance d_{LC} . It is obvious from Fig. 3 that the current version of that distance is far from optimal. Secondly,

and more importantly, classification of leaves also depends on their vein-structure, something which we completely neglected. These issues will be addressed in a forthcoming follow-up paper.

Acknowledgments

This work was partially supported by project NWO 613.002.056 *Computer-assisted identification of cetaceans* and by FP6 Network of Excellence MUSCLE.

References

1. Hillman, G., et al.: Computer-assisted photo-identification of flukes using blotch and scar patterns. In: Proceedings of 15th Biennial Conference on the Biology of Marine Mammals (December 2003)
2. Hu, M.K.: Visual pattern recognition by moment invariants. IRE Transactions on Information Theory IT-8, 179–187 (1962)
3. Mizroch, S., Beard, J., Lynde, M.: Computer Assisted Photo- Identification of Humpback Whales. In: Hammond, P., Mizroch, S., Donovan, G. (eds.) Individual Recognition of Cetaceans, pp. 63–70. International Whaling Commission, Cambridge (1990)
4. Oonincx, P.J., de Zeeuw, P.M.: Adaptive lifting for shape-based image retrieval. Pattern Recognition 36, 2663–2672 (2003)
5. Soille, P.: Morphological Image Analysis. Springer, Heidelberg (2003)
6. Ranguelova, E., Pauwels, E.J.: Saliency Detection and Matching Strategy for Photo-Identification of Humpback Whales. In: GVIP05. International Conference on Graphics, Vision and Image Processing, Cairo, Egypt, pp. 81–88 (December 2005)
7. Van Tienhoven, A., den Hartog, J., Reijns, R., Peddemors, V.: A computer-aided program for pattern-matching of natural marks on the spotted raggedtooth shark *carcharias taurus*. Journal of Applied Ecology 44(2), 273–280 (2007)
8. de Zeeuw, P.M.: A toolbox for the lifting scheme on quincunx grids (lisq). CWI Report PNA-R0224, Centrum voor Wiskunde en Informatica, Amsterdam (2002)

Distributed Generative Data Mining

Ruy Ramos and Rui Camacho

LIACC, Rua de Ceuta 118 - 6° 4050-190 Porto, Portugal

FEUP, Rua Dr Roberto Frias, 4200-465 Porto, Portugal

Abstract. A process of Knowledge Discovery in Databases (KDD) involving large amounts of data requires a considerable amount of computational power. The process may be done on a dedicated and expensive machinery or, for some tasks, one can use distributed computing techniques on a network of affordable machines. In either approach it is usual the user to specify the *workflow* of the sub-tasks composing the whole KDD process before execution starts.

In this paper we propose a technique that we call *Distributed Generative Data Mining*. The *generative* feature of the technique is due to its capability of generating new sub-tasks of the Data Mining analysis process at execution time. The *workflow* of sub-tasks of the DM is, therefore, dynamic.

To deploy the proposed technique we extended the Distributed Data Mining system HARVARD and adapted an Inductive Logic Programming system (IndLog) used in a Relational Data Mining task.

As a proof-of-concept, the extended system was used to analyse an artificial dataset of a credit scoring problem with eighty million records.

Keywords: Data Mining, Parallel and Distributed Computing, Inductive Logic Programming.

1 Introduction

As a result of more complex and efficient data acquisition tools and processes there is, in almost all, organisations huge amounts of data stored. Large amounts of money are invested in designing efficient data warehouses to store the collected data. This is happening not only in Science but mainly in industry. Analysis of such amounts of data has to be done using (semi-)automatic data analysis tools. Existing OLAP techniques are adequate for relatively simple analysis but completely inadequate for in-depth analysis of data. The discipline of Knowledge Discovery in Databases (KDD) is a valuable set of techniques to extract valuable information from large amounts of data (data warehouses). However, KDD [1] is facing nowadays two major problems. The amounts of data are so large that it is impractical or too costly to download the data into a single machine to analyse it. Also, due to the amounts of data or to its distributed nature in large corporations, it is the case that the data is spread across several physically

separated databases. These two problems prompted for a new area of research called Distributed and Parallel Data Mining [2]. This new area addresses the problem of analysing distributed databases and/or making the analysis in a distributed computing setting.

There are parallel versions of Decision Trees algorithms [3], parallel Association Rules Algorithms [4] and parallel Inductive Logic Programming (ILP) algorithms [5,6] that may be used in the (Relational) Data Mining step of KDD. To handle the large amounts of data these algorithms use a data partition approach and distribute the analysis work over a cluster of machines. One weakness of these systems is that if one of the machines fails the whole process has to be restarted, which is a serious drawback for long task's execution time. The implementation of such parallel DM algorithms are not designed with fault-tolerant features.

In this paper we propose a solution to make parallel DM algorithms such as Decision Trees, Association Rules or Inductive Logic Programming to be fault-tolerant and able to run on conventional PCs without disturbing the normal workings of an organisation. For that purpose we have extended the HARVARD [7] system to accommodate such desirable features for the parallel DM algorithms. The HARVARD system (**HARV**esting **AR**chitecture of idle machines **foR** **D**ata mining) has been developed as a computational distributed system capable of extracting knowledge from (very) large amounts of data using techniques of Data Mining (DM) namely Machine Learning (ML) algorithms. We take advantage of its following features. In a Condor [20] fashion, the system only assigns tasks to idle resources in the organisation. The system may access data distributed among several physically separated databases. The system is *independent* of the data analysis (ML) tool. It has very good facilities to recover from both slave and master nodes failure.

To integrate a parallel algorithm such as parallel Decision Trees, parallel Association Rules or parallel ILP we included in the HARVARD system the possibility of creating new tasks at run time at the HARVARD's task description level. This is the new *generative* feature of the distributed computing characteristic of HARVARD. The parallel DM algorithm on the other hand has to be adapted to be able to suggest new tasks in the HARVARD task description language. We present in this paper an example for the Inductive Logic Programming IndLog [9].

With this proposal we may include fault-tolerant features in some of the most popular distributed DM algorithms.

The rest of the paper is organised as follows. In the Section [2] we present the HARVARD system. In Section [3] we present basic notions of Inductive Logic Programming enough for the reader to understand the coupling of the analysis tool with the HARVARD system. We explain the generative technique for Data Mining in Section [4]. The deployment of the HARVARD system extension is described in Section [5]. Section [6] compares other projects with features close to HARVARD capabilities. We conclude in Section [7].

2 The HARVARD System

2.1 The Architecture

The KDD process is composed of a set of tasks that are executed according to a workflow plan. Both the tasks description and the workflow plan are provided by the user in two files. One file contains the tasks description and the second one contains the workflow. The workflow is specified using a control description language. Each task specification is made using XML. The information concerning each individual task include the name and location of the tool used in the task, the location of the data being processed in the task and the computational resources required (platform type, memory and disc requirements).

The distributed architecture of the HARVARD system is composed of a Master node and a set of computing nodes called Client (or Slave) nodes. The Master node is responsible for the control and scheduling the sub-tasks of the whole KDD process. Each Slave node executes application (sub-)tasks assigned by the Master node. Each node is composed by four modules that execute specific tasks to make the overall system working.

In what follows we refer to Figure 1 for the modular structure of both the Master and the Slave nodes. We now describe in detail the each node type.

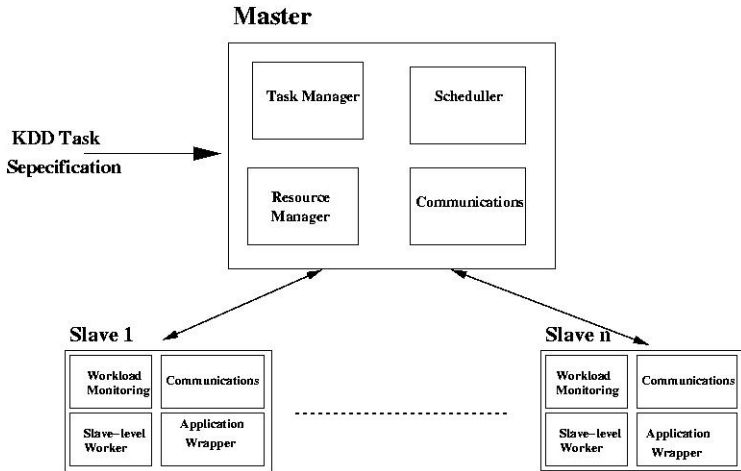


Fig. 1. The Harvard basic system architecture

The Master Node. The Master node is responsible for reading the KDD process specification and executing it. Each task of the KDD process is handled by the system as a **Working Unit (WU)**. Each WU will be assigned to a one or more machines. The assignment of a WU to more than one machine makes the system more tolerant to faults. It occurs when there are idle machines available and the task is expected to have long running times. There are other

fault tolerant features that we will refer below. When a WU finishes the results is associated with that WU and the status of the workflow graph updated. When the graph is completely traversed, meaning that the KDD process has finished, the result is returned to the user.

The Master node is composed by four modules: the Task manager; the Scheduler; the Resource Manager and; the Communications module.

The Task Manager Module. The basic function of the **Task Manager (TM)** module is to store and maintain and provide information concerning the tasks of the KDD process. The TM module constructs a graph structure representing the workflow of tasks.

It first reads and stores the specifications of all tasks composing the KDD process and then reads the workflow plan of the tasks and constructs a workflow graph structure. This module updates the status of the tasks in the graph and associates the results of each one when finished. At the end informs the user of the results of the KDD process. It may also be used to monitor the whole KDD process providing the user with information about the task finished, running and waiting computational resources to run.

The TM interacts with the Scheduler module. Looking at the workflow graph this module informs the Scheduler of ready to process tasks, provides a complete specification of each task and receives information concerning the terminations and results of each task.

The Resources Manager Module. The **Resources Manager (RM)** module stores and periodically updates information concerning the computational resources usable by the HARVARD system. When the system starts this module loads from a database the static information concerning all the computational resources usable by the system. That information is dynamically updated during the system execution. The information of each resource includes the type of platform and CPU, the amount of memory and disc space and a time-table with the periods the machine may be used. The workload of each machine is communicated periodically to this module to update so the system has a updated view of the resources. The RM module has a method (a match maker) to compute the “best” computational resource for a given request from the Scheduler. Each computation resources has a time-table of availability of the resource and the policy of use. This information state when the machine is available and in what conditions. The usage conditions may indicate that the system may use the machine only when there are no users logged in or by specifying a workload threshold that must be respected at all times.

The Task Manager module receives, from the Scheduler, requests for available machines satisfying a set of resources requirements and returns a best match at the moment. This module alerts the Scheduler that a task must be reassigned in two situations: if a machine is severely delayed to notify the TM module of its workload and; if the pre-established period of use of the machine is expired¹. The TM module receives periodically the workload of all running machines.

¹ In this case the task running on the machine is terminated.

The Communications Module. The **Communications (COM)** module is the only channel to access the world outside a node. All messages or requests concerning components or resources outside the node are processed by the COM module. This module exists in both Master and Slave nodes. To accomplish that task it implements several communication protocols that includes: RMI, socket, HTTP and JDBC. All these allows a client to download the task required software (HTTP), download the data (JDBC), send messages to the Master (sockets or RMI) and allows the Master to send messages to the Slaves (socket or RMI). It also allows the Master to keep a DB backup of its status and activities (JDBC) to allow a full recover in case of fault.

This Master COM module interacts via RMI or sockets with the COM module of the Slave to send messages. In a Master node the messages to be sent are received from the Scheduler module or the Resources Manager module. The former sends messages concerning task assignments and control directives whereas the later sends tasks status updated to be stored in a DB (fault tolerant purposes). The COM module receives and redirects the workload messages for the RM module. Received messages concerning tasks results are redirected to the Scheduler module.

The Scheduler Module. The **Scheduler** module controls the execution of the tasks composing the KDD process, launching, rescheduling or stopping the Work Units. The scheduler may also decide to assign a WU² to more than one Slave node. The scheduler inspects the workflow graph where the tasks interconnections and status are represented to decide what tasks to activate and when.

The Scheduler asks the Resource Manager module for the best match machine satisfying a given Work Unit requirements. With the results of such request the Scheduler assigns that WU to the given Slave and notifies the Slave via the Communications module. Whenever there is a change in the status of a WU the Scheduler is informed by the Task Manager of that event and triggers the (re)scheduling a new task.

A Slave Node. A Slave node does the actual data analysis work by running the Data Mining tool. In order to have a distributed system that is independent of the Data Mining tool the DM tool is involved in a wrapper that directly controls the DM tool. Each Slave also reports periodically its workload to the Resource Manager module of the Master. It is through the Slave's Communications module that the Slave downloads the DM tool and the data to be processed, and stores the results of the local analysis.

Each Slave has four modules: the Workload Monitoring (WM); the Worker (SW); the Application Wrapper (AW) and; the Communications (COM) module.

The Worker Module. The WU message is interpreted in this module. A WU usually results in several steps to be performed. A typical WU for analysing

² The ones considered more critical for some reason like training longer execution times.

data involves the downloading of the analysis tool, the download of the data, the processing and the return of the results. The Worker module controls all these steps by asking the Communications module to access the software and data and triggering the Application Wrapper module to execute the analysis. Finally it sends (via Communications module) the results to the Master.

The Worker nodes interacts with the Communication modules by sending it request to download the software and data and to return the final result. It also asks the Application Wrapper to start the analysis task and collects the results from it.

The Application Wrapper Module. The AW module completely controls the DM tool. It supplies the DM tool input stream and collects whatever appears at the DM output stream. Through the input stream the module provides the commands for the DM tool. The commands are provided in a file specified in the Working Unit specification. The output stream is stored in a file as the results file. The results file is uploaded to a database entry as specified in the WU specification. For the time being all the analysis of the results files are done in other follow up WU where special scripts written by the user do the necessary analysis. This permits the system to be independent of the DM tool.

The Workload Monitoring Module. This module monitors periodically the workload of the machine it is running and reports that information to the Resources Manager module of the Master. It also detects in a user has login into the machine. In that later case the Master is informed that the task running will be terminated. The Slaves enters a idle state where it just waits for the machine to be idle again.

Communications Module. The slave Communicating module is the only channel to the outside world. It has capabilities to download software using HTTP or ftp protocol, it may download data from a DB using JDBC and it can send and receive messages to and from the Master using RMI or sockets.

The Communications module interacts with all modules of the Slave node delivering and receiving messages.

2.2 Sub-tasks Workflow Description Language

The HARVARD system accepts as input a file describing the workflow of the sub-tasks composing the KDD process. The workflow is a graph with two kinds of nodes: sequential nodes and; parallel nodes. Each node stores a set of tasks to be executed or edges to other nodes. In a sequential node the set has an order and that order represents the sequential execution of the sub-tasks that must be respected. In a parallel node the tasks in the set may start all at the same time. In a parallel node there may be a *barrier* specifying the tasks that must terminate before the “execution” of the node is considered terminated. The graph has a root node where the tasks of the KDD process start executing.

Some of the steps in a KDD process are done quite frequent and most often are the same for a vast number of domains. For example feature subset selection or the DM tool parameter tuning are quite frequent pre-processing tasks in the KDD process. For these frequent tasks the task description language provides a set of macros for these complex operations. For example: to do a top-down feature selection up to two attributes or tune the “p” parameter using the values p1, p2 and p3. The system will then “unfold” those macros into the corresponding graph structure.

3 ILP in a *Tiny Nutshell*

Inductive Logic Programming (ILP) [10,11] is a discipline in the intersection of Machine Learning and Logic Programming. ILP studies techniques to (automatically) induce models for data. In ILP both the given data and the induced models are represented using First Order Logic. The data provided to an ILP system are of two kinds: i) examples and; ii) background knowledge. Examples are instances of the concept whose definition the system will induce. The background knowledge is any information the user thinks is relevant to construct the model. In the most popular ILP systems the induction process is mapped into a search on the space of all possible hypotheses (the hypothesis space). The system then uses any if well-know search algorithms to search the hypothesis space and return *the best* hypothesis according some specified criterion.

Some advantages of using ILP in DM tasks are the following. It has a very powerful description language to encode the constructed models. The user may give the system easily any information he considers relevant to produce the model. The background knowledge may include not only relation’s definitions but also numerical computations like regression models, geometric or statistical models etc. that are nicely combined in the final model. Most often ILP induced models are comprehensible to the user.

A major shortcoming of ILP systems is its efficiency. One possible approach the overcome ILP’s lack of efficiency is through the use of parallelism.

As described in [6] there are several approaches to parallelise an ILP system. One of the most simple but providing good results (see [5] for a comparison of several methods) consists in establishing a partition of the data and apply an ILP system to each subset of the data. The models induced by each ILP system on each subset are sent to a master node that combines the models. If the assembled model “explains” all the examples then the process stops and the model is the final model. Otherwise there are a next round of the same procedure where the examples not “explained” are subject a learning process equal to the first step. We have implemented in IndLog this approach to parallel ILP. It is a similar approach to the one described in [2] where classifiers are constructed for each set of the data partition and then sent to a central node where they are combined. In the ILP case this process is repeated until all examples are “explained”.

4 Distributed Generative Data Mining

The main cycle of a DM algorithm, such as Decision Trees, Association Rules or ILP is run a number of times that depend, among other things, on the input data. The number of nodes in a Decision Tree is not fixed before the algorithm that constructs the tree is run. The number of clauses in that and ILP algorithm induces is also not fixed before the algorithm processes the data. When developing parallel versions of these algorithms the number of tasks on each run will depend on the dataset being processed. On the other hand in a KDD process the number of tasks and their workflow is established by the user, using a tool like YALE [12] for example, and stays unchanged during the execution of the KDD process. To be able to integrate a parallel version of a DM algorithm in an automatic KDD setting one has to allow the tool to have a dynamic workflow of the tasks. For example new tasks are generated whenever a parallel Decision Tree is constructed and splits a new node. To accommodate such a situation we extended the HARVARD system to be able to dynamically accept new tasks during run time. New tasks are encoded as new nodes in the graph that represents the task's workflow.

In the data parallel version of IndLog a master node decides the split of the examples and assigns each slave node a subset. Instead of assigning directly the "sub-tasks" to the slaves via MPI interface, the IndLog translates the sub-tasks requests to a format that the Master node (Task Manager module) of the HARVARD understands. The HARVARD scheduler then assigns each task to existing idle machines. If there are too many idle machines then the same task may be run (redundantly) in more than one machine. In case of failure of one of the machines the analysis completes without delay.

The main advantages of the generative technique proposed in this paper is to increase the fault-tolerance of the analysis tool and to take full advantage of the HARVARD system. Using the HARVARD system the analysis does not disturb the normal workings of the organisation, does not require dedicated machines and has tolerance to failure of both the Master node and any of the Slave nodes.

5 Deployment of the HARVARD System

Just to test the feasibility of our approach and not to compare the systems performance on a specific problem we produced a large artificial data set with realistic information. The data set is on the domain of credit scoring. We characterised each instance with 55 attributes that correspond to the 55 fields of an actual form used by a real bank. The information used to generate the records was based on census information publicly available at the Brazilian national statistics office. We used a bank expert to provide the rules that assign the class value to the generated registers. The data set has 80 million registers. The data was stored in three MySQL databases in different machines. We used a laboratory with 15 PCs where Master students have classes and use for developing their practical works. The machines have dual boot so sometimes they start with Linux and sometimes with Windows. It takes several hours to analyse the data set on the reported computational environment using the HARVARD system with IndLog.

To analyse the data using IndLog [13,9] we had to provide scripts for the Application Wrapper of the Client node (see Figure 1) to control the IndLog system. The IndLog system was run in a data parallel fashion allowing a maximum of 50000 examples at each node.

To evaluate the fault-tolerant features we deliberately generated failures in some machines during the analysis process and under two circumstances. First we disconnect a machine running a task that was also assigned to other machine and notice no increase in the analysis time. Secondly we disconnected a machine where a task was running that was not assigned to any other machine. With a small overhead that task was reassigned to another machine and the analysis continued then normally.

6 Related Work

Our system is designed to take advantage of idle computers in an organisation and adequate for problems that may be decomposed into coarse grain sub-tasks. We present some related projects that can reach this objective but of differentiated form our architecture.

The Globus Alliance [14] is an international collaboration that does research in grid computing that seeks to enable "the construction of computational grids providing pervasive, dependable, and consistent access to high-performance computational resources, despite geographical distribution of both resources and users". One of the results of this research is "The Globus Toolkit" that is a "bag of services" like: resource allocation manager that provides creation, monitoring and management services; security infrastructure; monitoring and discovery services. For each of their services there is a programming interface in programming language C. These core services are used by other organisations and Globus to make high level components and systems [15,14,16].

The Boinc (Berkeley Open Infrastructure for Network Computing) [3] is a platform that makes it easy for scientists to create and operate public-resource computing projects. Workstations connected to the Internet by phone or DSL line can participate some project and share its own power computer to solve scientific problem when device is idle. The process is very simple, people interested to participate just installing a software client that connect a project master server. So, when workstation is idle some tasks may be executing. Some projects like SETI@home, Folding@home using the Boinc platform [17].

The Knowledge Grid is a specialised architecture in data mining tools that uses basic global services from Globus architecture [18]. The architecture design for Knowledge Grid following some principles: data heterogeneity and large data sets handling; algorithm integration and independence; compatibility with grid infrastructure and grid awareness; openness, scalability, security and data privacy [19].

Condor operates in workstation environment. The system aims to maximise the utilisation of workstation with as little interference as possible between

³ University of California - Berkeley- <http://boinc.berkeley.edu/>

the jobs is schedules and the activities of the people who own workstations. "Condor is specialised job and resource management system for compute intensive jobs. Like other full-featured systems, Condor provides a job management mechanism, scheduling policy, priority scheme, resource monitoring and resource management. Users submit their jobs to Condor when and where to run the based upon policy, monitors their progress, and ultimately informs the user upon completion" [20]. Condor allows almost any application that can run without user interaction to be managed. This is different from systems like Set@home and Protein Folding@home. These programs are custom written. Source code does not have to be modified in anyway to take advantage of these benefits. Code that can be re-linked with the Condor libraries gain two further abilities: the jobs can produce check-points and they can perform remote system calls [20].

Like the project Boinc, our architecture intends to use of idle workstations and also the system considers the heterogeneous environment with different operational systems (Linux, Windows, OS-X) but instead of it has a light client installed in each workstation it uses the Java Virtual Machine. A new approach will be present that to use old applications for data mining without re-build or re-compile with a new libraries like Condor or another approaches.

Besides, our proposal implements two-level language. A specific semantics for the administration of the data mining process, and other for specification of tasks of distributed processing. While a language is destined to the user for the definition of the process of the knowledge discovery, the other language is used by the system to manage the distributed processing.

7 Conclusions

We have made a proposal to run parallel Data Mining algorithms, such as parallel decision Trees, parallel Association Rules and parallel Inductive Logic Programming systems in a fault-tolerant setting. To achieve the desired fault-tolerant features we have extended the HARVARD system and have adapted the IndLog ILP system. The HARVARD system was extended with the possibility of including at run-time new tasks to schedule. In the ILP system a new module was encoded to communicate with the HARVARD Master node to suggest new tasks in the format of the task description language the the HARVARD system recognises.

The extended HARVARD system and the Inductive Logic Programming system IndLog were used to analyse an eighty million credit scoring dataset. The fault-tolerant features proved useful when simulating several failures on the machines during the analysis process.

References

1. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan-Kaufmann Publishers, San Francisco (2001)
2. Kargupta, H., Chan, P.: Advances in Distributed and Parallel Knowledge Discovery. AAAI/MIT Press, Cambridge (2000)

3. Amado, N., Gama, J., Silva, F.M.A.: Parallel Implementation of Decision Tree Learning Algorithms. In: Brazdil, P.B., Jorge, A.M. (eds.) EPIA 2001. LNCS (LNAI), vol. 2258, pp. 6–13. Springer, Heidelberg (2001)
4. Agrawal, R., Shafer, J.C.: Parallel mining of association rules. *IEEE Trans. On Knowledge And Data Engineering* 8, 962–969 (1996)
5. Fonseca, N.A., Silva, F., Camacho, R.: Strategies to Parallelize ILP Systems. In: Kramer, S., Pfahringer, B. (eds.) ILP 2005. LNCS (LNAI), vol. 3625, Springer, Heidelberg (2005)
6. Fonseca, N.A.: Parallelism in Inductive Logic Programming Systems. University of Porto, Porto (2006)
7. Ramos, R., Camacho, R., Souto, P.: A commodity platform for Distributed Data Mining – the HARVARD System. In: Perner, P. (ed.) ICDM 2006. Springer, Heidelberg (2006)
8. Litzkow, M.J., Livny, M., Mutka, M.W.: Condor—A Hunter of Idle Workstations. In: Proceedings of the 8th International Conference on Distributed Computing Systems, San Jose, California, pp. 104–111 (1988)
9. Camacho, R.: IndLog - Induction in Logic. In: Alferes, J.J., Leite, J.A. (eds.) JELIA 2004. LNCS (LNAI), vol. 3229, pp. 718–721. Springer, Heidelberg (2004)
10. Muggleton, S.: Inductive Logic Programming. *New Generation Computing* 8, 295–318 (1991)
11. Muggleton, S., De Raedt, L.: Inductive Logic Programming: Theory and Methods. *Journal of Logic Programming* 19/20, 629–679 (1994)
12. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: YALE: rapid prototyping for complex data mining tasks. In: KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 935–940 (2006)
13. Camacho, R.: Inducing Models of Human Control Skills using Machine Learning Algorithms. PhD thesis, Faculty of Engineering, University of Porto, Porto - Portugal (2000)
14. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure, pp. 259–278. Morgan-Kaufmann Publishers, San Francisco (1999)
15. Foster, I., Kesselman, C.: Globus: A Metacomputing Infrastructure Toolkit. *International Journal of Supercomputer Applications* 11(2), 115–128 (1997)
16. Foster, I.T., Kesselman, C., Tuecke, S.: The Anatomy of the Grid - Enabling Scalable Virtual Organizations. *CoRR*, vol. cs.AR/0103025 (2001)
17. Anderson, D.P.: BOINC: A System for Public-Resource Computing and Storage. In: David, P. (ed.) Proceedings on Fifth IEEE/ACM International Workshop on Grid Computing, pp. 4–10 (2004)
18. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure, pp. 259–278. Morgan-Kaufmann Publishers, San Francisco (1999)
19. Cannataro, M., Talia, D.: The Knowledge Grid. *Communications of the ACM* 46(1), 89–93 (2003)
20. Litzkow, M.J., Livny, M., Mutka, M.W.: Condor—A Hunter of Idle Workstations. In: Proceedings of the 8th International Conference on Distributed Computing Systems, San Jose, California, pp. 104–111 (1988)

Privacy-Preserving Discovery of Frequent Patterns in Time Series

Josenildo Costa da Silva and Matthias Klusch

German Research Center for Artificial Intelligence
Deduction and Multiagent Systems
Stuhlsatzenhausweg 3, 66123 Saarbruecken, Germany
{jcsilva,klusch}@dfki.de

Abstract. We present DPD-HE, a privacy preserving algorithm for mining time series data. We assume data is split among several sites. The problem is to find all frequent subsequences of time series without revealing local data to any site. Our solution exploit density estimate and secure multiparty computation techniques to provide privacy to a given extent.

1 Introduction

Frequent patterns discovery is an important step in many data mining algorithms, such as classification and clustering. Informally, a time series pattern is a subsequence that presents a given property and that occurs at different locations in the time series. In this paper we address the problem of finding *frequent, unknown* patterns given time series data split among different sites.

An important aspect we consider in this work is privacy of ownership and local data values. We assume that the sites are not willing to disclose exact values of original time series. Moreover, sites do not want other sites tracking any information to a specific site.

To solve this problem we introduce a density-based algorithm, which identify the most frequent subsequences occurring in the data set, considering the union of the data sets. The main idea is to represent subsequences of time series as points in a multidimensional space and compute the data density in this space. Using the additive property of density estimates, we produce a global density out of local ones. The pattern discovery problem is reduced to identifying local maxima in the density space. We show that our approach does protect the privacy of: (i) exact value of time series data; and (ii) identity of sites owning the data.

In the following we discuss related work (section 2) and show how we approach this problem (section 3). After that we present results and discussion of our experiments (section 4). Finally, we conclude and discuss future work (section 5).

2 Related Work

Pattern discovery problem has been extensively studied in bio-informatics, where the goal is to find frequent patterns in sequences of symbols, e.g. microarray data

analysis [1]. Recently this problem was extended to handle real-valued data [2]. More formally the problem is to identify the k -most frequent pattern occurring in the time series. Many approaches to PDTS problem have been proposed. Mörchen and Ultsch [3], for example, proposes using a grammar-based approach and Kadous [4] suggests clustering subsequences of the original time series to find prototypical shapes. These approaches, however, have some disadvantages. The grammar-based approach is too dependent on the knowledge of the possible patterns to be discovered and the clustering-based approach has been shown to be very problematic [5]. Liu et al. [6] proposed effective heuristics to solve PDTS problem defining patterns size m as multidimensional points in \mathbb{R}^m . Our work extends the original setting by adding distribution and privacy issues.

A related problem is sequence mining [7,8] where the time points are not equally spaced. The goal is to find temporal rules like “if event A happens then event B will happen after t units of time with $c\%$ confidence”. In our work we focus on equally spaced time points.

Works on privacy preserving data mining follow three main approaches. *Sanitization*, aims to modify the dataset such that sensitive patterns cannot be inferred. It was developed primarily to association rule mining (cf. [9,10]). The second approach is *data distortion*, in which the true value of any individual record is modified while keeping “global” properties of the data (cf. [11,12] among others). Finally, *SMC-based* approaches apply techniques from secure multi-party computation (SMC), which offers an assortment of basic tools for allowing multiple parties to jointly compute a function on their inputs while learning nothing except the result of the function (cf. [13,14]). In a SMC problem we are given a distributed network with each party holding secret inputs. The objective is to compute a function with the secret inputs ensuring that no party learns anything but the output. The general SMC problem was investigated by Goldreich et. al [15]. Latter, Lindell and Pinkas showed that privacy-preserving data mining problems could be solved using techniques of SMC [16]. Many applications of SMC to data mining have been proposed so far (cf. [17], [18], to name a few).

3 Density-Based Pattern Discovery in Time Series

In this section we present our approach to pattern discovery. First we describe our solution assuming centralized dataset. After that, we extend the ideas to the distributed case.

The following notation is used throughout this paper. Let $f : \mathbb{N} \rightarrow \mathbb{R}$ be a function from time stamps to reals. We define a time series $T = \{x_t = f(t) | 1 \leq t < m\}$ as ordered sequence of real numbers x coming from the measurement function f . The ordering is with respect to the time stamp t . A subsequence of T is denoted $\langle x_t, \dots, x_{t+v} \rangle$, for given integers $1 \leq t < m$ and $1 \leq v < m - t$. A frequent pattern in time series is a subsequence of the time series that reoccurs at different points of T .

3.1 Centralized Case

The pattern discovery problem we are addressing is defined as follows. Given a real-valued time series T , an integer k , to find the k -most frequent patterns occurring in T .

A brute force algorithm to solve this problem needs $O(|T|^2)$ comparisons, where $|T|$ represents the size of T . A more interesting approach follows a three-step scheme (cf. [19,2]):

1. Dimension reduction of the original time series T ;
2. Discretization of the reduced time series T into a sequence of symbols S over a given finite alphabet Σ ;
3. Pattern discovery algorithm using the symbol sequence S from the previous step.

This general scheme has the advantage of handling the high dimensionality of data, do not assume any knowledge on the structure of patterns and finally, provides a clear separation on the tree different tasks at hand. Our contribution here is a new strategy to perform the last step.

Dimension Reduction. Given a time series T and an integer n , reduce the dimensionality of T by averaging subsequences size n . Actually, this operation (proposed elsewhere [20]) is known as piecewise aggregate approximation (PAA):

$$\bar{x}_j = \frac{1}{n} \sum_{t=n(j-1)+1}^{nj} x_t \quad (1)$$

Discretization. Given a reduced time series from the previous step, compute the string S by substituting an element \bar{x}_j by a correspondent symbol σ in a given finite alphabet Σ . This is achieved by choosing break points $\{\beta_a\}$, $1 \leq a < |\Sigma| + 1$, such that each occurrence of a given value \bar{x}_j has the same probability [2]. Finally, the substitution rule is applied:

$$s_j = \begin{cases} \sigma_a & \text{iff } \beta_{a-1} \leq \bar{x}_j < \beta_a, 1 \leq a \leq |\Sigma| \\ \sigma_{|\Sigma|} & \text{otherwise} \end{cases} \quad (2)$$

To make sure this equation works we additionally requires that $\beta_1 = -\infty$ and $\beta_{|\Sigma|+1} = +\infty$

Discovery. The final part of the algorithm consists of estimating the density of subsequences σ of S and searching for those subsequences σ which have high density. The idea is to reduce the search for frequent subsequences to the search for dense regions in the pattern space. If we take subsequences of S of fixed size w , the pattern space is Σ^w .

A general approach to compute data density function is kernel-based density estimation. For a given kernel function \mathcal{K} such that $\int_{-\infty}^{+\infty} \mathcal{K}dx = 1$, an estimate of the true density is given by:

$$\hat{\varphi}(x) = \frac{1}{Nh} \sum_{x_i \in Neigh(x)} \mathcal{K} \left(\frac{D(x, x_i)}{h} \right) \tag{3}$$

where N is the total number of points, $D()$ is a distance function, h is a bandwidth parameter and $Neigh(x)$ is the neighborhood of point x (including the point x). We use the triangle kernel $\mathcal{K} = (1 - (\frac{x-x_i}{h})) I(\frac{x-x_i}{h} \leq 1)$, where I is the indicator function. We choose this kernel for its simplicity, but any other kernel can be used instead. The radius r is the bandwidth parameter. D is the Euclidean distance assuming the alphabet has a total order. An arbitrary points is denoted σ . Moreover, for a given string S , there are $\frac{|S|}{w}$ subsequences to consider. Finally, the estimate of the density of subsequences is:

$$\hat{\varphi}(\sigma) = \frac{w}{|S|r} \sum_{\sigma_i \in Neigh(\sigma)} \left(1 - \left(\frac{D(\sigma, \sigma_i)}{r} \right) \right) I \left(\frac{D(\sigma, \sigma_i)}{r} \leq 1 \right) \tag{4}$$

Local maxima in pattern space correspond to strings that reoccur more frequently than others. The set of frequent patterns \mathcal{P} is a set of reoccurring strings, each of them representing a local maxima.

$$\mathcal{P} = \{ \sigma \in \Sigma^w : \forall \sigma' \in \Sigma^w (|\sigma - \sigma'| \leq r \rightarrow \varphi(\sigma) > \varphi(\sigma')) \} \tag{5}$$

The parameter r works as a radius assuring that the center of density region is a good descriptor of frequent subsequences inside the ball radius r . This constraint reduces the number of pattern taking only the most representatives ones.

To find local maxima quickly we store a representation of each point in the pattern space together with its density estimates. This structure is ordered by density. When a set of candidates is chosen we just weed out the pattern which are to similar according to the radius r , as already explained

The space complexity can be very high for large values of w . In practice, however, only a few variations of all possible patterns appear. Consequently, we can exploit this scarcity to improve the density estimate storage costs.

Now we extend the pattern discovery problem to the distributed case.

3.2 Distributed Case

The problem of discovering pattern in distributed time series can be described as follows. Given an integer k , and a set of sites $\mathcal{L} = \{L_i\}_{1 \leq i \leq P}$, each of them with a local time series T_i , the problem is to find the set \mathcal{P} of the k -most frequent patterns occurring in $T = \bigcup_{i=1}^P T_i$, such that:

1. The total communication cost is minimized
2. The result using the distributed data T_i is the same if the algorithm runs using $T = \bigcup_{i=1}^P T_i$

The key observation here is that the density estimate is additive. Therefore we can compute local density and sum them up to produce the global estimates. With the global density estimate we can perform the discovery step locally using the ideas discussed in centralized case.

We assume that the time series data collected at different sites refers to the same variable and has the same time spacing. We also assume that the sites negotiate on the parameters k, n, w, Σ, r . If the negotiation fails the protocol stops. If an agreement is found, then they can proceed.

3.3 Addressing Privacy

In a distributed environment, there is always the threat that an eavesdropper is listening in on the conversation among the sites. To avoid this threat we use a secure multiparty computation technique: homomorphic encryption.

Homomorphic Encryption (HE) scheme allows for parties to perform arithmetical operation directly without decryption. Here we are using Paillier scheme [21] which is an additive homomorphic. So, given two messages m_1 and m_2 the following holds: $E(m_1) \cdot E(m_2) = E(m_1 + m_2)$. Paillier scheme consists of the following steps:

Key Generation. Let $N = pq$ be a RSA modulus and g be an integer of order $\alpha N \bmod N^2$, for some integer α . The public key is (N, g) and the private key is $\lambda(N) = lcm((p - 1), (q - 1))$.

Encryption. The encryption of message $m \in \mathbb{Z}_N$ is $E(m) = g^m r^N \bmod N^2$, with r randomly selected from \mathbb{Z}_N

Decryption. Given a cipher text c the message is computed as follows:

$$m = \frac{L(c^{\lambda(N)} \bmod N^2)}{L(g^{\lambda(N)} \bmod N^2)}$$

where $L(u) = \frac{u-1}{N}$.

Now, we show how we put all together in the DPD-HE algorithm.

3.4 DPD-HE Algorithm

DPD-HE is our algorithm based on the ideas discussed in the previous sections. Its main phases are:

Phase 1: preparation. The mining group \mathcal{L} agrees on the parameters and each party computes the local density of the words generated from the local time series data. The initiator, which is the peer that proposes and coordinates the mining session, create a key pair and publicize its public key.

Phase 2: computing global estimates. Each party encrypts its local density estimate using the public key from the initiator. Then after it receives the encrypted partial sum from its neighbor, sums its local encrypted density estimate, and then sends it to the next neighbor.

Phase 3: termination. When all parties added its local encrypted density estimate the last party sent the encrypted sum to the initiator. The initiator

decrypts it, adds its own local density estimates, and finally, searches the local maxima in the global density estimate. The result is sent back to the mining group encrypted with the public key of each party.

Algorithm 1. Initiator

Input: $k, T_i, n, w, \Sigma, \mathcal{L}, r;$

Output: $\mathcal{P};$

At the initiator do:

- 1: $(n, w, \Sigma) \leftarrow \text{negotiateParameters}(\mathcal{L});$
 - 2: $\mathcal{H}_1 \leftarrow \text{createsHashTable}(T_1, n, w, \Sigma);$
 - 3: $LDE_1 \leftarrow \mathcal{H}_1.\text{density}(\theta, T_1);$
 - 4: $(PK, SK) \leftarrow \text{generateKeyPairs}();$
 - 5: $\text{broadcast}(\mathcal{L}, PK);$
 - 6: $EGDE_{|\mathcal{L}|-1} \leftarrow \text{receive}(L_{|\mathcal{L}|-1}, \text{Encr}(PK, \sum_{j=1}^{|\mathcal{L}|-1} LDE_j));$
 - 7: $GDE \leftarrow \text{Decr}(SK, EGDE_{|\mathcal{L}|-1}) + LDE_1;$
 - 8: $\mathcal{P} \leftarrow GDE.\text{findDensityCenters}(k, r);$
 - 9: **for** $i = 1$ **to** $|\mathcal{L}|$ **do**
 - 10: $\text{broadcast}(\mathcal{L}, \text{Encr}(PK_i, \mathcal{P}));$
 - 11: **end for**
-

Algorithm 2. Arbitrary Party

Input: $k, T_i, n, w, \Sigma, \mathcal{L}, r;$

Output: $\mathcal{P};$

At an arbitrary party j do:

- 1: $(n, w, \Sigma) \leftarrow \text{negotiateParameters}(\mathcal{L});$
 - 2: $\mathcal{H}_j \leftarrow \text{createsHashTable}(T_j, n, w, \Sigma);$
 - 3: $LDE_j \leftarrow \mathcal{H}_j.\text{density}(\theta, T_j);$
 - 4: $PK \leftarrow \text{receive}(L_1);$
 - 5: $EGDE_{j-1} \leftarrow \text{receive}(L_{j-1});$
 - 6: $\text{send}(L_{j+1}, \text{Encr}(PK, LDE_j) + EGDE_{j-1});$
 - 7: $\mathcal{P} \leftarrow \text{Decr}(SK_j, \text{receive}(L_1));$
-

DPD-HE takes $O(|T|)$ steps, where $|T|$ means the size of the original time series, mainly due the reduction and discretization steps, which requires a pass through the entire dataset. On the other hand it requires $O(|\Sigma|^w)$ space, which can be controlled by the value of w . There are only 2 rounds of messages, one of which informs the mining results. Each message has size $O(|\Sigma|^w)$, for given globals w and Σ . The communication costs may be controlled by choosing the value of w .

Let us now discuss the privacy properties of the proposed approach. In general there are two main attack scenarios in a distributed data mining application. The first scenario is an attack from an insider, who may acts like a normal member of the mining group. The second scenario is an attack from an outsider. In this case the attacker listens into the conversation and tries to learn information from the eavesdropped messages.

In this paper we focus on the first scenario. By using the homomorphic encryption we can expect that an outside attack won't be successful with high probability. The critical situation is when the outsider has a partner inside the group. But this reduces the problem to an inside scenario. Therefore, we concentrate here on how much privacy is provided against an insider attack.

We assume that the attacker know all parameters, for it is an insider. Now, let us define the privacy of a time series as the accuracy of reconstructed data an attacker may produce. This is along with the information-theoretical definition of privacy proposed elsewhere [22]. The privacy of a given random variable is proportional to its uncertainty, which translates to entropy in the information theory.

Definition 1. *Given a random variable Y with domain Ω_Y and probability density function $p(y)$ its privacy \mathbf{PR} is given by*

$$\mathbf{PR}(Y) = 2^{h(Y)} \tag{6}$$

where $h(Y)$ is the differential entropy of Y with $h(Y) = - \int_{\Omega_Y} p(y) \log_2 p(y) dy$.

One interpretation of this definition is that the privacy of a given variable A is the length of the interval where the attacker knows it is located with probability 1, which is $2^{h(A)}$. The larger the interval from the point of view of the attacker, the more privacy we have. The idea is that the discretization step in DPD-HE provides a given amount of uncertainty. We want to use the information on the discretization transform to compute the uncertainty and consequently the privacy of each point in the original time series X . The following theorem captures this idea.

Theorem 1. *Let Σ be an alphabet of symbols used by the DPD-HE protocol. Let T be a time series and $S \in \Sigma^w$ be its transform according to the discretization step. Let $\{\beta_j \in \mathbb{R}\}_{j=1}^{|\Sigma|+1}$ be a set of breakpoints which divides the domain of the time series T in $|\Sigma|$ equiprobable regions. For a given point x_t if its transformed counterpart $s_j = \sigma_a$ is known, than its privacy level is given by:*

$$\mathbf{PR}(x_t) = |\beta_{a+1} - \beta_a| \tag{7}$$

Proof. This is a consequence of the discretization step. Since we know that the symbol σ_a comes from the substitution rule in the discretization step, we know that average \bar{x}_j of the points in the subsequence $\langle x_t, \dots, x_{t+n} \rangle$ lies in the interval (β_a, β_{a+1}) . In the absence of further information, the only suitable option is to model \bar{x}_j as a random variable uniformly distributed in the given interval, i.e. $\bar{x}_j \sim U(\beta_a, \beta_{a+1})$. Now, using the privacy metric in Eq. (7) we have:

$$\begin{aligned} \mathbf{PR}(x_t) &= 2^{h(x_t)} = 2^{\int_{\beta_a}^{\beta_{a+1}} p(x) \log_2 p(x) dx} \\ &= 2^{\log_2(\beta_{a+1} - \beta_a)} = |\beta_{a+1} - \beta_a| \end{aligned}$$

As a consequence, the more symbols in the alphabet, the less privacy we get, which is according to the intuition, as the discretized version tends to get the "shape" of the original data.

Theorem 2. *Assuming no collusion among attackers, DPD-HE keeps the privacy about ownership of a given the local density estimate.*

Proof. (Sketch) The overall security is a consequence of the security of the Paillier encryption scheme, which was shown to be semantically secure elsewhere [21]. Since no local density is decrypted at any site but the initiator, assuming that the initiator do not collude, we have that an attacker cannot assign any of information a specific site.

4 Experiments

In the following we present some results of our experiments. We implemented our approach in GNU Octave, which is a high-level language for numerical computation.

In the experiments reported here we used the *power data* records the electricity consumption from Netherlands Energy Research Foundation (ECN) for one year, recorded every 15 minutes. There are 35 040 data points corresponding to the year of 1997. Figure 1 shows an excerpt of the power data. This data set has a pattern structure that can be observed visually.

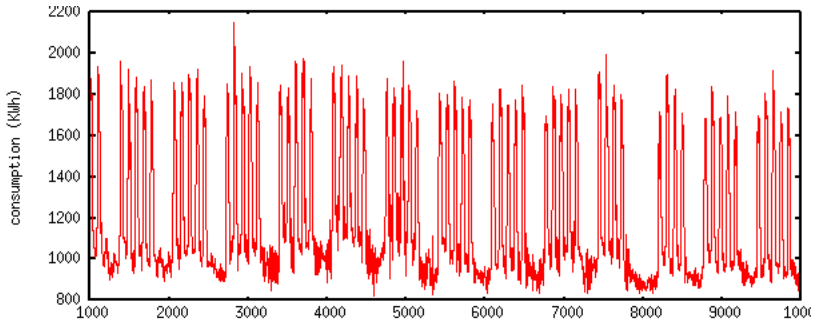


Fig. 1. Excerpt of power data

We set the parameter as follows. Subsequence size $n = 96$, which corresponds to one day (with one measurement every 15 minutes). We choose pattern size $w = 7$ for it represents a week. The alphabet Σ was set to $\{a, b, c, d\}$. Symbol ‘a’ represents lowest values and ‘d’ represents highest value of consumption. The radius was set to $r = 1$. Larger values of r produces larger neighborhoods. The density landscape becomes smoother which may reduce the number of local maxima and consequently the number of patterns. Choosing smaller values of r produces a more spiky density with more local maxima and more patterns. Therefore, r help us to control the number of patterns. With these parameters values we found 2 frequent patterns. The first pattern is “ccccaa” which corresponds to a normal week. The second pattern is “acccaa” which correspond to weeks having a holiday on the first day. Figure 2 shows one instance of each pattern.

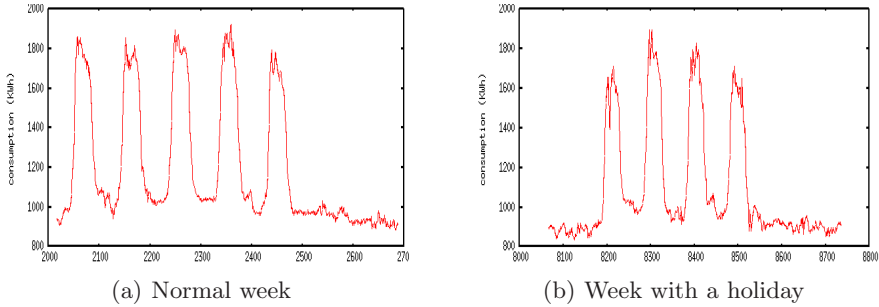


Fig. 2. (a) An instance of a *normal week*, the most frequent subsequence in the power data, showing high consumption on work days and low consumption at weekends. (b) An instance of a *week with holiday* pattern in the power data. In this example, the Monday was a holiday.

Results with different alphabet sizes, with all other parameters set as above, found an increasing number of patterns. This is mainly because larger alphabets produce more accurate discretization, what allows for a more detailed differentiation among the patterns. These additional patterns basically presented refinements of the more general pattern “five day high + two days low”.

Results of performance shown in figure 3(a). We performed the experiments with the same parameters values as in the previous experiments. For the performance tests we created a synthetic time series with 300 000 points, by cloning power data 10 times. As shown in the figure 3(a), the CPU time increases linearly with the size of the time series.

Figure 3(b) shows the results of privacy vs. size of alphabet. To measure the privacy we used the interval size corresponding to each alphabet symbol in the discretization step. In the initial case, with just one symbol, we used the size of interval from the minimum to the maximum value observed in the time series, which is 1 056 KWh. Assuming that max and min values are public, the

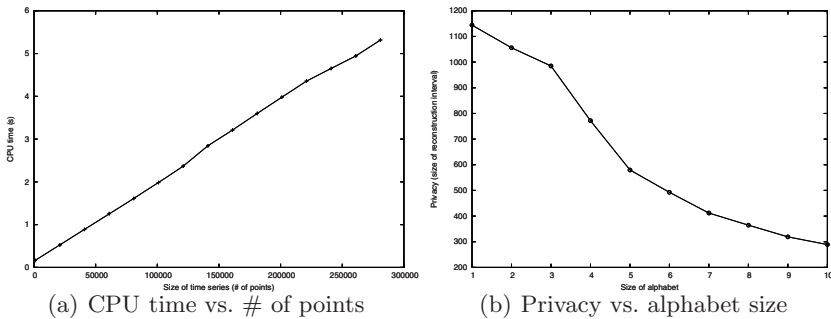


Fig. 3. (a) Time performance with increasing size of time series; (b) Privacy level with increasing size of alphabet

attacker can compute the entropy of a random variable X over this interval, and consequently the privacy level $2^h(X)$. That is the privacy we get when the sequence consists of symbols from a singleton alphabet. In the figure we see the decrease of privacy by using more symbols to discretize the time series. With 10 symbols we get a privacy level of 300 KWh, which means that an attacker cannot reconstruct a data point within an interval smaller than 300 KWh. It is up to the user, however, to decide whether or not a given privacy level is enough.

5 Conclusion

We presented the DPD-HE, an algorithm for pattern discovery in distributed time series. It is time efficient (linear in the size of time series) and provides privacy of ownership and data values. The main idea is to use density estimation to identify the most frequent subsequences. The additive of density estimates allow us to extend the basic idea to the distributed case. By using a cryptographic protocol, it is possible to compute the global density without disclosing local data. Experiments on real and synthetic data sets showed that most of frequent patterns can be found, and no false negative is produced.

Future work includes an application to distributed time series clustering. We also have plans to improve the privacy level in scenarios where the attackers form collusion. In this case we may use permutation on the secure sum protocol. Another interesting extension is to work with multivariate data, which calls for a more efficient space usage and investigation on possible data leakage through correlation, which is typically very high in time series data.

Acknowledgments

The authors are very grateful to the “UCR Time Series Data Mining Archive” [23] for providing us the with time series data. The authors also thank German Ministry of Education and Research for support through grant BMBF 01-IW-D02-SCALLOPS and the Brazilian Ministry for Education for support through grant CAPES 0791/024.

References

1. Jensen, K.L., Styczynski, M.P., Rigoutsos, I., Stephanopoulos, G.N.: A generic motif discovery algorithm for sequential data. *Bioinformatics* 22, 21–28 (2006)
2. Lin, J., Keogh, E., Lonardi, S., Patel, P.: Finding motifs in time series. In: Proc. of the Second Workshop on Temporal Data Mining, Edmonton, Alberta, Canada (July 2002)
3. Moerchen, F., Utsch, A.: Discovering temporal knowledge in multivariate time series. In: Proc. GfKI 04, Dortmund, Germany (2004)
4. Kadous, M.W.: Learning comprehensible descriptions of multivariate time series. In: Proc. 16th International Conf. on Machine Learning, pp. 454–463. Morgan Kaufmann, San Francisco (1999)
5. Lin, J., Keogh, E., Truppel, W.: Clustering of streaming time series is meaningless. In: Proc. of the 8th ACM DMKD, San Diego, California, pp. 56–65. ACM Press, New York (2003)

6. Liu, Z., Yu, J.X., Lin, X., Lu, H., Wang, W.: Locating motifs in time-series datas. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 343–353. Springer, Heidelberg (2005)
7. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Yu, P.S., Chen, A.S.P. (eds.) Eleventh International Conference on Data Engineering, Taipei, Taiwan, pp. 3–14. IEEE Computer Society Press, Los Alamitos (1995)
8. Srikant, R., Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) EDBT. Proc. 5th Int. Conf. Extending Database Technology, 25–29 1996, vol. 1057, pp. 3–17. Springer, Heidelberg (1996)
9. Atallah, M., Bertino, E., Elmagarmid, A., Ibrahim, M., Verykios, V.: Disclosure limitation of sensitive rules. In: KDEX'99. Proceedings of 1999 IEEE Knowledge and Data Engineering Exchange Workshop, Chicago, IL, November 1999, pp. 45–52. IEEE Computer Society Press, Los Alamitos (1999)
10. Saygin, Y., Verykios, V.S., Elmagarmid, A.K.: Privacy preserving association rule mining. In: Research Issues in Data Engineering (RIDE) (2002)
11. Evfimievski, A., Srikant, R., Agrawal, R., Gehrke, J.: Privacy preserving mining of association rules. In: Proceedings of 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD), Edomonton, Alberta, Canada, ACM Press, New York (2002)
12. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: Proc. of the ACM SIGMOD Conference on Management of Data, Dallas, Texas, May 2000, pp. 439–450. ACM Press, New York (2000)
13. Pinkas, B.: Cryptographic techniques for privacy-preserving data mining. ACM SIGKDD Explorations Newsletter 4(2), 12–19 (2002)
14. Vaidya, J., Clifton, C.: Secure set intesection cardinality with application to association rule mining, Submitted to ACM Transactions on Information and Systems Security (March 2003)
15. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game. In: In Proc. of the 19th annual ACM conference on Theory of computing, pp. 218–229. ACM Press, New York (1987)
16. Lindell, Y., Pinkas, B.: Privacy preserving data mining. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 36–54. Springer, Heidelberg (2000)
17. Du, W., Zhan, Z.: Building Decision Tree Classifier on Private Data. In: IEEE ICDM Workshop on Privacy, Security and Data Mining, Maebashi City, Japan. CRPIT, vol. 14, pp. 1–8. IEEE Computer Society Press, Los Alamitos (2002)
18. Kantarcioglu, M., Vaidya, J.: Privacy preserving naive bayes classifier for horizontally pertitioned data. In: IEEE ICDM Workshop on Privacy Preserving Data Mining, November 2003, pp. 3–9. IEEE Computer Society Press, Los Alamitos (2003)
19. Tanaka, Y., Iwamoto, K., Uehara, K.: Discovery of time-series motif from multi-dimensional data based on mdl principle. Machine Learning 58, 269–300 (2005)
20. Keogh, E.J., Chakrabarti, K., Pazzani, M.J., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Knowledge and Information Systems 3(3), 263–286 (2000)
21. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, p. 223. Springer, Heidelberg (1999)
22. Agrawal, D., Aggarwal, C.C.: On the design and quantification of privacy preserving data mining algorithms. In: 20th ACM PODS, Santa Barbara, California, May 2001, pp. 247–255. ACM Press, New York (2001)
23. Keogh, E., Folias, T.: The ucr time series data mining archive (2002), <http://www.cs.ucr.edu/~eamonn/TSDMA/index.html>

Efficient Non Linear Time Series Prediction Using Non Linear Signal Analysis and Neural Networks in Chaotic Diode Resonator Circuits

M.P. Hanias and D.A. Karras

Chalkis Institute of Technology, Greece, Automation Dept., Psachna, Evoia,
Hellas (Greece) P.C. 34400
dakarras@teiha.gr, dakarras@ieee.org, mhanias@teiha.gr

Abstract. A novel non linear signal prediction method is presented using non linear signal analysis and deterministic chaos techniques in combination with neural networks for a diode resonator chaotic circuit. Multisim is used to simulate the circuit and show the presence of chaos. The Time series analysis is performed by the method proposed by Grasberger and Procaccia, involving estimation of the correlation and minimum embedding dimension as well as of the corresponding Kolmogorov entropy. These parameters are used to construct the first stage of a one step / multistep predictor while a back-propagation Artificial Neural Network (ANN) is involved in the second stage to enhance prediction results. The novelty of the proposed two stage predictor lies on that the backpropagation ANN is employed as a second order predictor, that is as an error predictor of the non-linear signal analysis stage application. This novel two stage predictor is evaluated through an extensive experimental study.

Keywords: prediction, non-linear signal analysis, diode, chaos, time series, correlation dimension, prediction, neural networks.

1 Introduction

Time series forecasting, or time series prediction, takes an existing series of data and forecasts the future data values. The goal is to observe or model the existing data series to enable future unknown data values to be forecasted accurately.

A novel two-stage time series prediction method is presented in this paper and is applied to the prediction of a chaotic signal produced by a diode resonator chaotic circuit. This circuit, being quite simple, illustrates how chaos can be generated. We have selected Multisim [1] to simulate circuits since it provides an interface as close as possible to the real implementation environment. In addition, complete circuits implementation and oscilloscope graphical plots are all presented. While non-linear signal analysis methods have been quite extensively studied and applied in several systems presenting chaos, chaotic time series prediction for electronic circuits is a field not too deeply investigated so far. Chaos has already been recognized to be present in electronic circuits [2]-[5]. Some preliminary investigations on such time series prediction have been performed by the authors in [6]. The present paper aims at developing efficient predictors for such chaotic time series. To this end, the classical

nonlinear signal analysis (i.e [7]-[8]) has been involved as a first stage of the proposed predictor, while back-propagation neural networks have been employed in the second stage to enhance first stage results, being a second order predictor for the first time in then relevant literature. An extensive experimental study shows that the proposed predictor is very favourably evaluated in terms of accuracy with the classical nonlinear signal analysis methodology.

2 The Non Autonomous Driven RL Diode Circuit

A non autonomous chaotic circuit referred to as the driven RL-diode circuit (*RLD*) [2-4] shown in Fig 1.

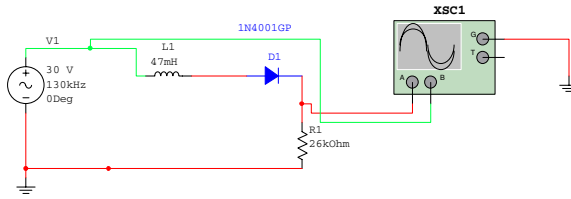


Fig. 1. RL-Diode chaotic circuit

It consists of a series connection of an ac-voltage source, a linear resistor R_1 , a linear inductor L_1 and a diode D_1 type 1N4001GP, that is the only nonlinear circuit element. An important feature of this circuit is that the current i (or the voltage across the resistor R) can be chaotic although the input voltage V_1 is nonchaotic. The usual procedure is to choose a parameter that strongly affects the system. We found that for $V_1=30V$ RMS and input frequency $f=130$ KHz, inductance $L_1=47mH$, the response is a chaotic one. The results of the Multisim simulation are shown in Fig. 2. The RL-diode was implemented and the voltage oscillations across the resistor V_{R1} and its phase portrait V_1 vs V_{R1} are shown in Fig.2.

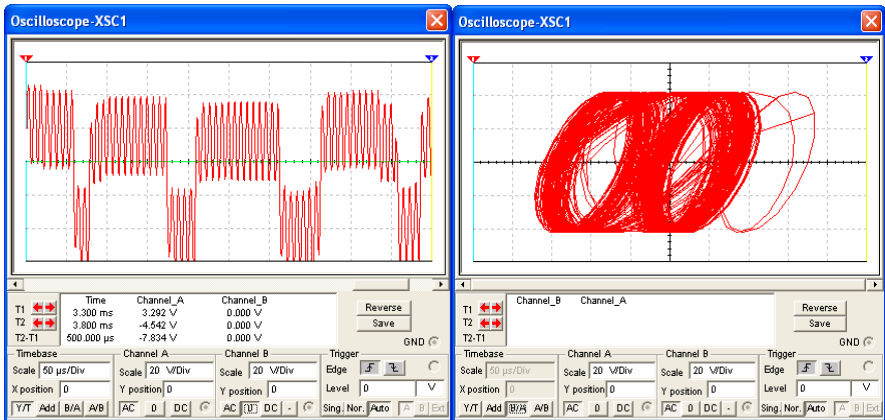


Fig. 2. Time series $V_{R1}(t)$ (left) Phase portrait of V_1 versus V_{R1} (right)

3 The Proposed Novel Prediction Methodology

3.1 First Stage: The Non Linear Signal Analysis Process

Time series prediction takes an existing series of data

$$x_{t-n}, \dots, x_{t-2}, x_{t-1}, x_t \tag{1}$$

and forecasts the future

$$x_{t+1}, x_{t+2}, \dots \tag{2}$$

data values. Taking into account this point of view we could interpret the data produced by the RLD circuit as a non-linear chaotic time series. The goal is to observe or model the existing data series to enable future unknown data values to be forecasted accurately.

To evaluate the resulted time series, the method proposed by Grasberger and Procaccia [7,8] and successfully applied in similar cases [9-11] has been applied in order to define the first stage of the proposed predictor. According to Takens theory [12] the measured time series were used to reconstruct the original phase space. For this purpose we calculated the correlation integral, for the simulated signal, defined by the following relation [13].

$$C_m(r) = \lim_{N \rightarrow \infty} \frac{2}{(N)(N+1)} \sum_{i=1}^N \sum_{j=i+1}^N H\left\{r - \left(\sum_{k=1}^m [x_{i+k} - x_{j+k}]^2\right)^{\frac{1}{2}}\right\} \tag{3}$$

for $\lim r \rightarrow \infty$, where

N.....is the number of points,

H.....is the Heaviside function,

m is the embedding dimension

In the above equation N is the number of the experimental points here N=16337, X_i is a point in the m dimensional phase space with X_i given by the following relation [12]

$$X_i = \{V_{RI}(t_i), V_{RI}(t_i + \tau), V_{RI}(t_i + 2\tau), \dots, V_{RI}(t_i + (m-1)\tau)\} \tag{4}$$

The vector

$X_i = \{V_{RI}(t_i), V_{RI}(t_i + \tau), V_{RI}(t_i + 2\tau), \dots, V_{RI}(t_i + (m-1)\tau)\}$, represents a point to the m dimensional phase space in which the attractor is embedded each time, where τ is the time delay $\tau = i\Delta t$ determined by the first minimum of the time delayed mutual information $I(\tau)$ [13-16]. In our case, because of sample rate $\Delta t = 4.8 \times 10^{-7}$ s, the mutual information function exhibits a local minimum at $\tau = 6$ time steps as shown at Fig .3.

We used this value for the reconstruction of phase space. With (3) dividing this space into hypercubes with a linear dimension r we count all points with mutual distance less than r. It has been proven [7-8] that if our attractor is a strange one, the correlation integral is proportional to r^v where v is a measure of the dimension of the attractor, called the correlation dimension. The correlation integral $C(r)$ has been numerically calculated as a function of r from formula (3), for embedding dimensions $m = 1..10$. In Fig 4 (upper insert) the slopes v of the lower linear parts of these double logarithmic curves give information characterizing the attractor.

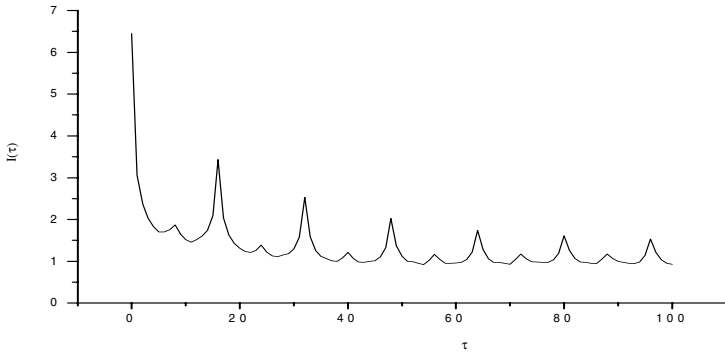


Fig. 3. Average Mutal Information vs time delay τ

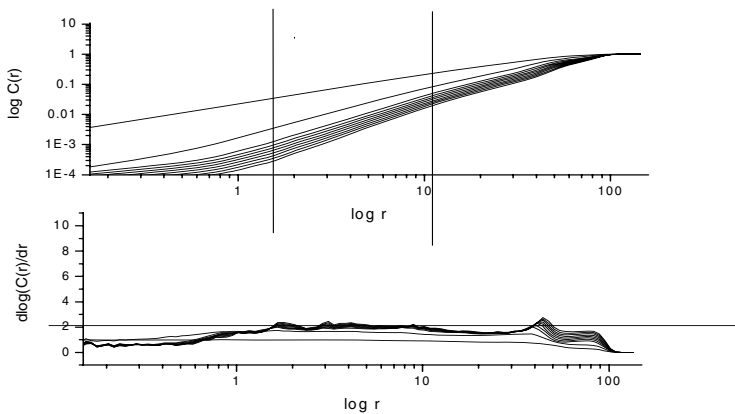


Fig. 4. The correlation intergral $C(r)$ vs $\log r$, for different embedding dimensions m (upper insert). The corresponding slopes and the scaling region (lower insert).

In fig 4 (lower insert) the corresponding average slopes v are given as a function of the embedding dimension m . It is obvious from these curves that v tends to saturate, for higher m 's, at non integer value $v=2.11$ with this value of v the minimum embedding dimension could be $m_{\min}=3$ [13]. So the minimum embedding dimension of the attractor for one to one embedding is 3.

In order to get more precise measurements of the strength of the chaos present in the oscillations we have introduced the Kolmogorov entropy. According to [13] the method described above also gives an estimate of the Kolmogorov entropy, i.e. the correlation integral $C(r)$ scales with the embedding dimension m according to the following relation

$$C(r) \sim e^{-m\tau K_2} \tag{5}$$

Where K_2 is a lower bound to the Kolomogorov entropy. From the plateau of fig 5 we estimate $K_2=0.11$ bit/s.

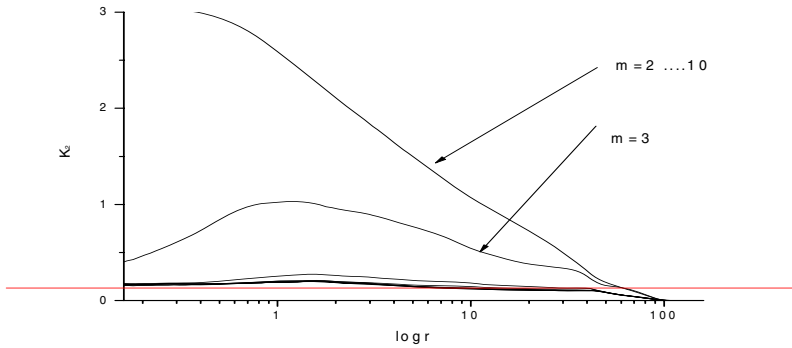


Fig. 5. The Kolmogorov entropy vs $\log r$ for different embedding dimensions

3.2 Second Stage: The Back-Propagation ANN as a Second Order Predictor

The proposed novel algorithm to enhance non-linear signal analysis prediction is as follows:

1. To predict point V_{i+1} , we determine the last known state of the system as represented by vector $X = [V_i, V_{i-\tau}, V_{i-2\tau}, V_{i-(m-1)\tau}]$, where m is the embedding dimension and τ is the time delay.
2. With optimum values of delay time and embedding dimension m we then search the time series to find k similar states that have occurred in the past, where “similarity” is determined by evaluating the distance between vector X and its neighbour vector X' in the m -dimensional state space. So k close states (usually nearest neighbours of X) of the system that have occurred in the past are found, by computing their distances from X .
3. We used a fixed size of nearest neighbours K (calculated for optimizing prediction performance in the training phase). if a state $X' = [V'_i, V'_{i-\tau}, V'_{i-2\tau}, V'_{i-(m-1)\tau}]$ in the neighbourhood of X resulted in the observation V'_{i+1} in the past, then the point V_{i+1} which we want to predict must be somewhere near V'_{i+1} . This is the main concept of nonlinear signal analysis of first order approximation.
4. It is reasonable to calculate $V_{i+1} = (\sum q_k V'_k) / \sum q_k$ where q_k the distance between current state X and neighboring state X_k , whereas V'_k the corresponding prediction from X'_k vector (from the training set). The above sum is considered for all X neighbors
5. Our proposition to enhance prediction results is to write down $V_{i+1} = (\sum q_k V'_k) / \sum q_k + \text{error}_{V_{i+1}}$, where $(\sum q_k V'_k) / \sum q_k$ is the first order prediction and $\text{error}_{V_{i+1}}$, is the prediction error to be minimized provided it is calculated properly. Therefore, it is a second order approximation proposal to predict such an error. This $\text{error}_{V_{i+1}}$, could be calculated through a suitable neural network as an error predictor. This is exactly the main concept of the proposed novel methodology.
6. Suppose err_k the corresponding prediction error measured through the above procedure for each neighboring state X'_k of given current state X above (out of the K neighbours of X). This err_k is known through the training set, since for each X'_k in the training set we can calculate its corresponding K neighbours from the training set, and then, estimate, using step 5 above, the associated

- err_k. Therefore, for k we construct all K such err_k. Then, we feed these K values as inputs to a back-propagation neural network of K-L1-L2-1 architecture. This network, trained with the conjugate gradient algorithm, due to the large training set, since it is known to be the best algorithm for large data sets and ANN architectures [18], should be able to predict state's X error error_{V_{i+1}}.
- The training set needed for step 6 is constructed for each state X of the training set by estimating all corresponding err_k of its K neighbours and its associated error_{V_{i+1}}, which of course serves as the desired output of the corresponding input pattern

4 Experimental Study

We have used a simulated time series from RLD circuit with V₁=30V RMS and input frequency f=130 KHz and we predict the voltage V across the resistor.

We use locally linear models to predict the one step and the multistep procedures. That is, instead of fitting one complex model with many coefficients to the entire data set, we fit many simple models (low order polynomials) to small portion of the data set depending on the geometry of the local neighborhood of the dynamical system. [17].The general procedure is the following: To predict point V_{i+1}, we determine the last known state of the system as represented by vector $\mathbf{X} = [V_i, V_{i-\tau}, V_{i-2\tau}, V_{i-(m-1)\tau}]$, where m is the embedding dimension and τ is the time delay.

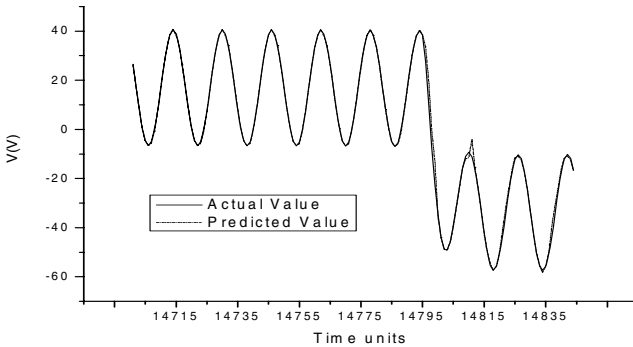


Fig. 6. One step prediction

So we use as a delay time the value of $\tau=6$ as before. From previous analysis the correlation dimension for RLD circuit is found $v=2.11$. With optimum values of delay time and embedding dimension $m=3$ we then search the time series to find k similar states that have occurred in the past, where “similarity” is determined by evaluating the distance between vector \mathbf{X} and its neighbour vector \mathbf{X}' in the m-dimensional state space. So k close states (usually nearest neighbours of \mathbf{X}) of the system that have occurred in the past are found, by computing their distances from \mathbf{X} as explained in section 3 above.

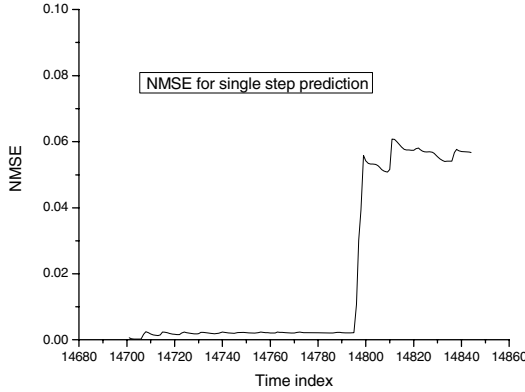


Fig. 7. Mean squared error of our predictor normalized by the mean squared error of the random walk predictor for one step prediction

The idea is to fit a map which extrapolates \mathbf{X} and its k nearest neighbours to determine the next value. If the observable signal was generated by some deterministic map $M(V_i, V_{i-\tau}, V_{i-2\tau}, \dots, V_{i-(m-1)\tau}) = V_{i+\tau}$, that map can be recovered (reconstructed) from the data by looking at its behaviour in the neighbourhood of \mathbf{X} . Using this map, an approximate value of V_{i+1} can be obtained. We used a fixed size of nearest neighbours $k=36$. Now we can use this map to predict V_{i+1} . In other words, we make an assumption that M is fairly smooth around \mathbf{X} , and so if a state $\mathbf{X}' = [V'_i, V'_{i-\tau}, V'_{i-2\tau}, \dots, V'_{i-(m-1)\tau}]$ in the neighbourhood of \mathbf{X} resulted in the observation V'_{i+1} in the past, then the point V_{i+1} which we want to predict must be somewhere near V'_{i+1} . [17]. We have employed both the one step and multistep ahead prediction methods. In the one step ahead prediction, after each step in the future is predicted, the actual value is utilized for the next one-step prediction. In contrast, the multistep prediction is based only on the initial k states.

The calculated performance is otherwise known as the Normalized Mean Squared Error (NMSE) is calculated by (5-1),

$$NMSE = MAX \left(\frac{\sum_{i=1}^{NP} (\tilde{V}_i - V_i)^2}{\sum_{i=1}^{NP} (\bar{V}_i - V_i)^2}, \frac{\sum_{i=1}^{NP} (\tilde{V}_i - V_i)^2}{\sum_{i=1}^{NP} (V_{i-1} - V_i)^2} \right) \quad (5-1),$$

where \tilde{V}_i is the predicted value, V_i , the actual value, \bar{V} is the average actual value, and NP is the range of values in the prediction interval.

From (5-1), it can be seen that NMSE is the mean squared error of our predictor normalized by the mean squared error a random walk predictor. By definition, the minimum value of NMSE is 0. At that value, there is the exact match between the actual and predicted values. The higher NMSE, the worse is our prediction as compared to the trivial predictors. If NMSE is equal to 1, our prediction is as good as the prediction by the trivial predictor. If NMSE is greater than 1, our prediction worsens. With values of $\tau=6, m=3$ we achieved the minimum NMSE.

The second stage back-propagation ANN is of a 36-60-60-1 architecture.

We used 14700 data points and predicted the evolution for 889 succeeding dimensionless time steps. The results are shown at fig 6 where the one step ahead predicted values are coming from prediction out-of-sample set, where we pretend that we know the data only up until this point, and we try to predict from there, while the one step ahead predicted values are coming from prediction out-of-sample set. The NMSE is shown at fig 7 for the one step prediction.

We use the same procedure as before but with multi-step ahead predictions. The results are shown at Fig - 8 The NMSE is shown at Fig - 9 for the multi step prediction.

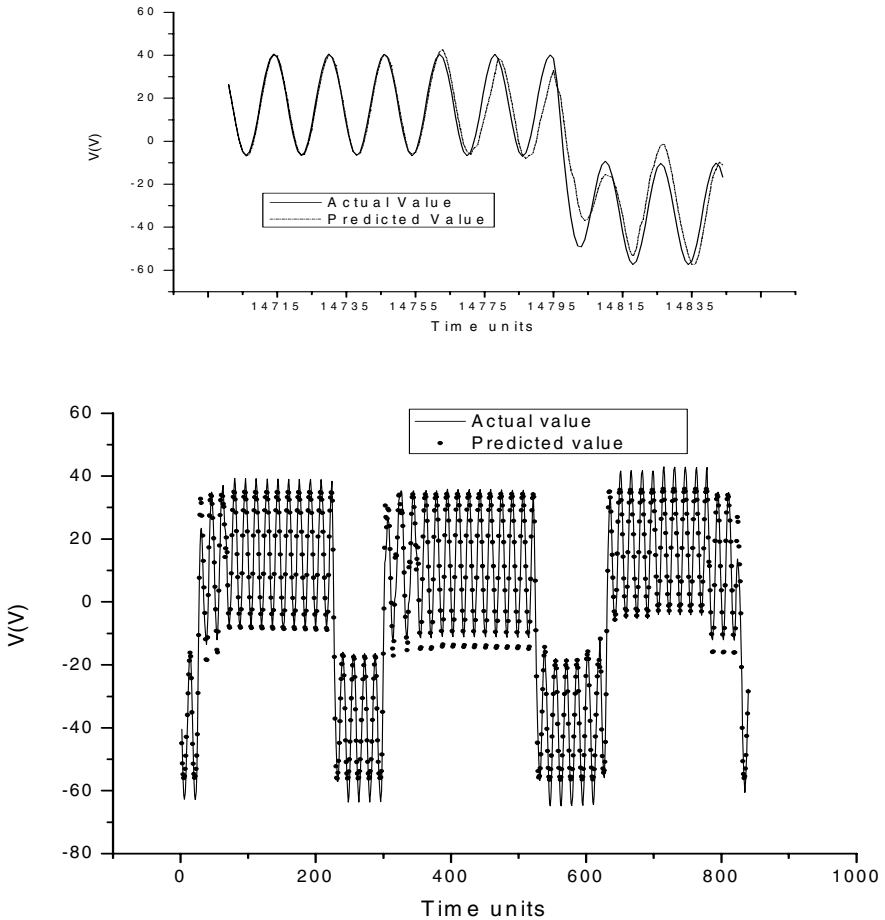


Fig. 8. Multistep prediction, Actual and predicted time series for 10 time steps ahead for the total set of points and the unknown time series (lower insert, in detail)

In comparison, when a first stage only predictor is used without the proposed neural network of stage 2, on average, for the 889 unknown data points we have achieved

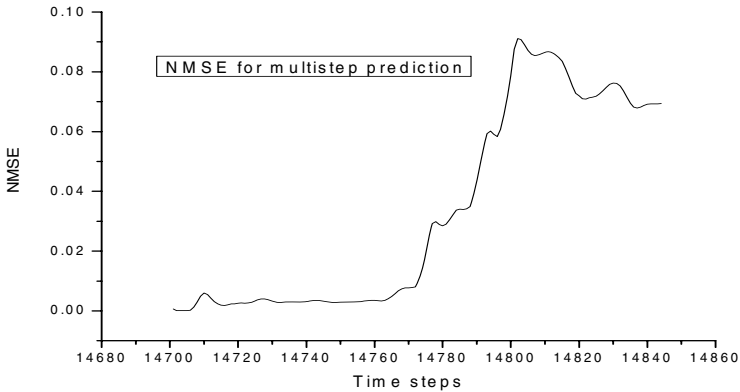


Fig.9. Mean squared error of our predictor normalized by the mean squared error of the random walk predictor for multistep prediction

8.5% worse performance in the one-step prediction for the NMSE and 7.8% worse performance in the multistep prediction experiments. Therefore, the proposed methodology is worth evaluating it further in larger scale experiments.

5 Conclusions and Future Trends

We have proposed a novel two-stage time series prediction scheme based on nonlinear signal analysis methods and a novel error prediction back propagation ANN trained with the conjugate gradient algorithm. Applying the methods of non linear analysis in the time series produced by the chaotic simple RLD circuit we found that the strange attractor that governs the phenomenon is a Lorenz type attractor with a correlation dimension $\nu=2.11$ who is stretching and folding in a 3 dimension phase space. This is also evident from the one step ahead and multistep successful predictions with the use of the correspondence strange attractor invariants as input parameters, and the efficient ANN model introduced in the second stage of the proposed predictor.

We believe that for a detailed understanding of chaos in the *RLD* circuits these results must be combined with the reverse-recovery effect and all of its nonlinearities. The proposed prediction methodology might be applied successfully in other chaotic time series too, since it is quite general. This is, also, a future target of the authors.

References

- [1] Lonngren, K.E.: Notes to accompany a student laboratory experiment on chaos. IEEE Transactions on Education 34(1) (1991)
- [2] Matsumato, T., Chua, L., Tanaka, S.: Simplest Chaotic Nonautonomous Circuit. Phys. Rev. A 30, 1155–1157 (1984)
- [3] Azzouz, A., Hasler, M.: Orbits of the R-L-Diode Circuit. IEEE Transaction on Circuits and Systems 37, 1330–1339 (1990)

- [4] Aissi, C.: Introducing chaotic circuits in an undergraduate electronic course. In: Proceedings of the 2002 ASEE Gulf-Southwest Annual Conference, The University of Louisiana at Lafayette, March 20-22, 2002. Copyright © 2002, American Society for Engineering Education (2002)
- [5] de Moraes, R.M., Anlage, S.M.: Unified model and reverse recovery nonlinearities of the driven diode resonator. *Phys. Rev. E* 68, 26–201 (2003)
- [6] Haniias, M.P., Giannaris, G., Spyridakis, A., Rigas, A.: Time series Analysis in chaotic diode resonator circuit. *Chaos Solitons & fractals* 27(2), 569–573 (2006)
- [7] Grassberger, P., Procaccia, I.: Characterization of strange attractors. *Phys. Rev. Lett.* 50, 346–349 (1983)
- [8] Grassberger, P., Procaccia, I.: Measuring the strangeness of strange attractors. *Physica D* 9, 189 (1983)
- [9] Haniias, M.P., Kalomiros, J.A., Karakotsou, C., Anagnostopoulos, A.N., Spyridelis, J.: Quasi-Periodic and Chaotic Self - Excited Voltage Oscillations in TlInTe₂. *Phys. Rev. B* 49, 16994 (1994)
- [10] Mozdy, E., Newell, T.C., Alsing, P.M., Kovanis, V., Gavrielides, A.: Synchronization and control in a unidirectionally coupled array of chaotic diode resonators. *Physical Review E* 51(6), 5371–5376 (1995)
- [11] Abarbanel, H.D.I.: *Analysis of Observed Chaotic Data*. Springer, New York (1996)
- [12] Takens, F.: *Lecture Notes in Mathematics* 898 (1981)
- [13] Kantz, H., Schreiber, T.: *Nonlinear Time Series Analysis*. Cambridge University Press, Cambridge (1997)
- [14] Aasen, T., Kugiumtzis, D., Nordahl, S.H.G.: Procedure for Estimating the Correlation Dimension of Optokinetic Nystagmus Signals. *Computers and Biomedical Research* 30, 95–116 (1997)
- [15] Fraser, A.M., Swinney, H.L.: Independent coordinates for strange attractors from mutual information. *Phys. Rev. A* 33, 1134–1140 (1986)
- [16] Fraser, A.M.: *IEEE transaction of information Theory* 35, 245 (1989)
- [17] Kononov, E.: *Virtual Recurrence Analysis*, Version 4.9 (2006), (email:eugenek@ix.net.com.com)
- [18] Haykin, S.: *Neural Networks, a comprehensive foundation*, 2nd edn. Prentice-Hall, Englewood Cliffs (1999)

Using Disjunctions in Association Mining

Martin Ralbovský¹ and Tomáš Kuchař²

¹ Department of Information and Knowledge Engineering,
University of Economics, Prague, W. Churchill Sq. 4, 130 67 Praha 3, Czech Republic
martin.ralbovsky@gmail.com

² Department of Software Engineering, Faculty of Mathematics and Physics
Charles University, Malostransk nm. 25, 118 01 Prague, Czech Republic
tomas.kuchar@gmail.com

Abstract. The paper focuses on usage of disjunction of items in association rules mining. We used the GUHA method instead of the traditional *apriori* algorithm and enhanced the former implementations of the method with ability of disjunctions setting between items. Experiments were conducted in our Ferda data mining environment on data from the medical domain. We found strong and meaningful association rules that could not be obtained without the usage of disjunction.

Keywords: Association Mining, Disjunction, GUHA Method, Ferda.

1 Introduction

Association rules mining is an important technique widely used in the KDD community [8]. Most of the tools nowadays use the *apriori* algorithm, or its modifications [1] [2]. The algorithm searches for frequent (or large) itemsets with given minimal *support* and then calculates *confidence*. We will refer to this algorithm as to classical association mining. Its authors considered only the conjunctions (and possibly negations) of items.

Yet sometimes it is feasible to examine disjunctions of items. Consider following example: Medical expert wants to find associations between beer consumption and other characteristics of a patient (blood pressure, level of cholesterol, body mass index...). The examined data contains information about consumption of three different types of beer: light 7 degree beer, drought 10 degree beer and lager 12 degree beer [1]. It is likely to happen that the number of patients drinking 7 degree *or* 12 degree beer is higher then the number of patients drinking 7 degree *and* 12 degree beer. More formally, from the rule $A \rightarrow B$ one can get rule $A \rightarrow B \vee C$ easily than the rule $A \rightarrow B \wedge C$. For semantically close entities [2]

¹ This categorization of beer is traditional in the Czech Republic and represents the weight percentage of mash in the end product. 7 degree beer contains about 2% of alcohol, 10 degree beer about 3 to 4% and 12 degree beer about 4 to 5% of alcohol.

² Drinking of different types of beer is semantically close characteristics of a patient.

one can therefore use disjunctions and mine for rules with higher support (and possibly other characteristics).

The aim of this paper is to present an enhancement of classical association mining with the possibility of disjunction setting between the items. One cannot use *a priori* for disjunctions, because the algorithm searches for frequent itemsets by binding items to already known itemsets (of length k) with conjunction to form itemsets (of length $k+1$). We used the older GUHA method instead, which mines for modifications of association rules. The generalization enables disjunctions between items and had several partial implementations before the personal computer era. We created a new implementation in our Ferda tool and conducted experiments with medical data using more strict requirements for rules than *support* and *confidence*. Meaningful rules have been found; these rules could not be found without disjunction usage and have interesting characteristics that should be subject of further research.

The paper is structured as follows: Section 2 explains the GUHA method and its relation to classical association mining. Section 3 states a brief history of tools implementing the GUHA method leading to our Ferda tool. Section 4 describes conducted experiments. Section 5 draws fields of further research and finally section 6 concludes the work.

2 Principles of Association Mining with GUHA

2.1 The GUHA Method

GUHA method is one of the first methods of exploratory data analysis, developed in the mid-sixties in Prague. It is a general mainframe for retrieving interesting knowledge from data. The method has firm theoretical foundations based on observational calculi and statistics [4], [5]. For purpose of this paper let us explain only the basic principles of the method, as shown in Figure 1.

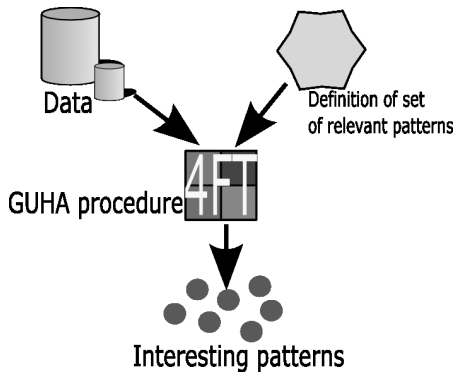


Fig. 1. The GUHA method

GUHA method is realized by GUHA procedures such as 4FT procedure to be described, located in the middle of the figure³. Inputs of the procedure are data and a simple definition of a possibly large set of relevant patterns, which will be discussed in detail in the following section^{2.2}. The procedure automatically generates all the relevant patterns and verifies them against the provided data. Patterns that are true are output of the procedure.

Although GUHA is not in principle restricted to mining association rules, the most used GUHA procedures mine for generalized association rules, as defined in ^{1.2}. Section^{2.3} introduces 4FT, procedure for association rules mining used in our work. Comparison study between the classical association mining and mining using GUHA can be found in ⁶.

2.2 Definition of Set of Relevant Patterns

This section shows how set of relevant patterns is defined in association rules mining with GUHA. We use the term attribute in the sense of *categorical attribute*, i.e. attribute with finite number of values.

Definition 1. Let A be an attribute, $A = \{a_1, a_2 \dots a_n\}$ and $\alpha \subset A$, $\alpha \neq \emptyset$. Then $A(\alpha)$ is a **basic Boolean attribute**.

Definition 2. Each basic Boolean attribute is a **Boolean attribute**. If α and β are Boolean attributes, $\alpha \wedge \beta$, $\alpha \vee \beta$ and $\neg \alpha$ are **Boolean attributes**.

The above stated definition was introduced in ^{1.2} when formalizing association rules. *Boolean attributes* are used as antecedents or succedents⁴ in GUHA procedures, as will be described in section^{2.3}. Our Ferda tool is the first program to enable full *Boolean attribute* definition including disjunction and recursion. Example 1 shows us creation of *Boolean attributes* from the beer consumption example from the introduction.

Example 1

The examined data includes three attributes: **beer7** = {no, low, high}, **beer10** = {no, low, high} and **beer12** = {no, low, high} for consumption of 7, 10 and 12 degree beer respectively.

Examples of *basic Boolean attributes* are **beer7(no)**, **beer10(no, low)** or **beer12(high)**⁵.

³ Even though we present only one GUHA procedure in this work, there are five more procedures working above one data table implemented in Ferda and also two relational procedures under development.

⁴ In classical association mining called consequents.

⁵ Obviously, not all the subsets of an attribute are meaningful to verify. Our method allows user to define special subsets such as subsets with a given length, intervals, cyclic intervals or cuts for ordinal data.

Table 1. 4FT contingency table

\underline{M}	ψ	$\neg\psi$
φ	a	b
$\neg\varphi$	c	d

Table 1: 4ft table

Then we combine *basic Boolean attributes* with logical operators to form a rule: **(beer7(no) \vee beer10(no, low)) \wedge \neg beer12(high)**, which is an example of *Boolean attribute*.

2.3 4FT Procedure

Classical association mining searches rules in form $X \longrightarrow Y$, where X and Y are sets of items. Procedure 4FT searches (in the simplified form) for rules in form $\varphi \approx \psi$, where φ and ψ are *Boolean attributes* and \approx is a *4ft-quantifier*⁶. Relation $\varphi \approx \psi$ is evaluated on the basis of *4ft table*, as shown in Table 1.

A *4ft table* is a quadruple of natural numbers $\langle a, b, c, d \rangle$ so that:

- a : number of objects (rows of M) satisfying φ and ψ
- b : number of objects (rows of M) satisfying φ and not satisfying ψ
- c : number of objects (rows of M) not satisfying φ but satisfying ψ
- d : number of objects (rows of M) satisfying neither φ nor ψ

4ft-quantifier expresses kind of dependency between φ and ψ . The quantifier is defined as a condition over the *4ft table*. By the expression **strict quantifier** we mean that there are no rules that satisfy the quantifier in the usual case. Occurrence of such quantifier means a very strong relation in the data. In the following sections we present three quantifiers used in our work, the *founded implication*, *double founded implication* and *founded equivalence* quantifiers⁷. This part of the paper was greatly inspired by [11], where detailed explanation of the most used quantifiers can be found.

2.4 Founded Implication Quantifier

The founded implication is basic quantifier for the 4FT procedure introduced in [4] and is defined by following condition:

$$a \leq Base \wedge \frac{a}{a + b} \geq p$$

Here $Base$ and p are threshold parameters of the procedure. The $Base$ parameter represents absolute number of objects that satisfies φ . In our work we will use

⁶ The more complex form includes another *Boolean attribute* as a condition. In our work we do not mine for conditional rules, therefore we omit the more complex definition.

⁷ There are many other quantifiers invented and implemented for the 4FT procedure.

relative *Base* representation, $\frac{a}{a+b+c+d}$. So, the *Base* parameter corresponds to the *support* and p to the *confidence* parameters of classical association mining. When using the 4FT procedure with *founded implication quantifier* and constructing *Boolean attributes* only with conjunctions, we get the same results as if using classical association mining. Quantifier can be verbally interpreted with the expression *tendency to*.

Example 2

Association rule *Patients that drink 12 degree beer tend be overweight* is an example of rule we can found with *founded implication*. This rule can be formally written as **beer12(high)** $\Rightarrow_{p,Base}$ **BMI(overweight)**, where $\Rightarrow_{p,Base}$ stands for *founded implication*.

2.5 Double Founded Implication Quantifier

The *double founded implication* quantifier enriches the *founded implication* with symmetry feature. Symmetry means that when $\varphi \approx \psi$ is valid, when $\psi \approx \varphi$ should be also valid. The quantifier has also threshold parameters *Base* and p and is defined by following condition:

$$a \leq Base \wedge \frac{a}{a+b+c} \geq p$$

Again, we will use the relative representation of *Base*, $\frac{a}{a+b+c+d}$. We consider *double founded implication* a *strict quantifier*. However, we wanted to use the quantifier in our experiments to question the possibilities of disjunctions. The most suitable verbal expression for the quantifier is *relation of equivalence*.

Example 3

The sign for *double founded implication* quantifier is $\Leftrightarrow_{p,Base}$. The rule **beer12 (high)** $\Leftrightarrow_{p,Base}$ **BMI(overweight)** with the *Boolean attributes* from example 2 can be verbally interpreted as *Drinking 12 degree beer is in relation of equivalence with being overweight among the observed patients*.

2.6 Founded Equivalence Quantifier

The last presented quantifier is the *founded equivalence*. It is a stronger quantifier than *founded implication* in terms of equivalence; ability of two entities to attain the same logical values. The condition for the quantifier is

$$a \leq Base \wedge \frac{a+d}{a+b+c+d} \geq p$$

For *founded equivalence*, we will also use the relative representation of *Base*, $\frac{a}{a+b+c+d}$. The fraction $\frac{a+d}{a+b+c+d}$ means the proportion of objects in the data matrix having φ and ψ both equal to 0 or 1, to all objects. Similarly, *Base* and p are threshold parameters for the quantifier. As well as *double founded implication*, the *founded equivalence* is considered to be a *strict quantifier*. Quantifier can be verbally interpreted as *equivalent occurrence*.

Example 4

The sign for the *founded equivalence* quantifier is $\equiv_{p,Base}$. Generalized association rule **beer12(high)** $\equiv_{p,Base}$ **BMI(overweight)** can be translated to verbal form as *Consumption of 12 degree beer and being overweight has equivalent occurrence among the observed patients.*

3 GUHA and Ferda

We would not achieve results presented in this work without 40 years long research of the GUHA method and development of tools that implemented individual GUHA procedures. This section acknowledges achievements made by researchers and developers in the past and briefly describes history that lead to the state-of-the-art Ferda tool. See [3] for more information about history of GUHA method.

The development of first GUHA procedure started in 1956. In modern terminology, it mined for association rules with given *confidence* with one item as a antecedent and one item as a consequent [3]. The results, published in [4], were clearly ahead of their time, long before terms like data mining or knowledge discovery from databases were invented.

First GUHA procedure to consider disjunctions was the IMPL procedure introduced in [5]. The procedure mined for rules in form $CONJ \Rightarrow DISJ$, where $CONJ$ and $DISJ$ are elementary conjunctions and disjunctions [8]. \Rightarrow is an *implicational quantifier* [9]. The implementation of the procedure [13] [15] used for the first time the bit string approach [10]. The input data were represented by strings of bits, which dramatically increased performance of the procedure.

The LISp-Miner tool [11] started in 1996 and contributed greatly to the level of contemporary GUHA tools by implementing six GUHA procedures and implementing coefficients - generation of special types of subsets of an attribute [14]. GUHA procedure 4ft-Miner implemented in LISp-Miner is predecessor of procedure 4FT introduced in this work. 4ft-Miner does not allow construction of *Boolean attributes*, it constructs *partial cedents* instead and until very recently it did not allow disjunctions. *Partial cedent* is a restricted non recursive *Boolean attribute*, more details are to be found in [14].

Ferda started as a student project to create a new visual environment for the LISp-Miner system [9]. In the first version, creators used the LISp-Miner GUHA procedures. The second version of the system, implemented in work [10], uses the bit string approach and enables full definition of *Boolean attribute*. It is the first modern tool (runs on personal computers) to implement disjunctions and recursion of *basic Boolean attributes*. The procedure 4FT implemented in Ferda

⁸ Elementary conjunction is a conjunction made from one element subsets of an attribute.

⁹ There are formal classes of *4ft-quantifiers* defined in [12]. *Implicational quantifiers* are one of the classes.

¹⁰ Also known as *granular computing*.

¹¹ See <http://lispminer.vse.cz>

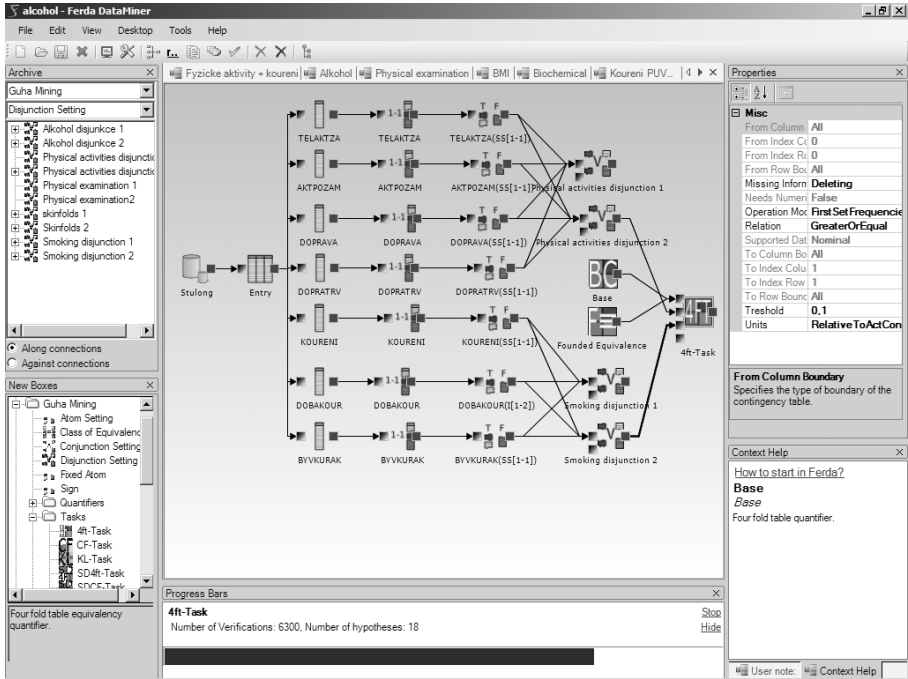


Fig. 2. Ferda environment

is the most generalized version of the original ASSOC procedure defined in [5]. The user environment is shown in Figure 2.

4 Experiments

In order to test the possibilities of disjunctions of items, we carried out experiment, which consisted of testing number of analytical questions. We chose the STULONG data set, introduced in section 4.1. The limitations and criteria that led to the experiment setup are described in sections 4.2 and 4.3. Performance is discussed in section 4.4. We found interesting and also some unexpected results which are summarized in section 4.5.

4.1 STULONG Data Set

We decided to use the STULONG medical data set for our experiments. The data set contains data about longitudinal study of atherosclerosis risk factors. There are two main reasons to choose this data set. First reason is that STULONG is relatively known among KDD researchers – it served as the examined data set for three ECML/PKDD Discovery challenges. There are meaningful analytical

questions defined on the data set to be examined, which is the second reason. In our experiments we wanted to answer these questions¹².

4.2 Limitations

Before explaining setup of the experiments, let us note two major limitations of mining disjunctions in general. These limitations affect our experiments as well. First limitation is generation of *non prime* rules. The rule is *prime* when it is true in the examined data and when the rule cannot be derived from a more simple rule also true in the examined data. More details can be found in [5]

Our implementation does not guarantee generation of prime rules. Therefore we expect the number of valid rules to rise dramatically when using disjunctions with quantifiers that are not *strict*. However, we may use disjunctions with *strict quantifiers*, where it is a common case that no rules are found at all.

The other limitation is interpretation of rules with disjunction. The motivation example of beer consumption in section 11 showed that it makes sense to use disjunctions with semantically close attributes, possibly synonyms or taxonomically bound attributes. Interpretation of disjunction of random attributes is rather problematic. Therefore we used for disjunctions only the attributes of the same attribute groups¹³.

4.3 Setup

From above stated limitations we concluded a setup for experiments. We answered 15 analytical questions concerning relations between significant characteristics of patients' entry examination¹⁴

1. *What are the relations between social factors and the following characteristics of men in the respective groups:*
 - (a) *Physical activity at work and in free time*
 - (b) *Smoking*
 - (c) *Alcohol consumption*

¹² EUROMISE: The STULONG Project <http://euromise.vse.cz/stulong>

The STULONG Project is partially supported by project no.LN00B107 of the Ministry of Education of the Czech Republic and by grant no.2003/23 of the Internal Grant Agency of the University of Economics, Prague. The STULONG study was carried out at the 2nd Department of Medicine, 1st Faculty of Medicine of Charles University and Charles University Hospital (head Prof. M. Aschermann, MD, SDr, FESC), under the supervision of Prof. F. Boudík, MD, ScD, with collaboration of M. Tomečková, MD, PhD and Ass. Prof. J. Bultas, MD, PhD. The data were transferred to electronic form by the European Centre of Medical Informatics, Statistics and Epidemiology of Charles University and Academy of Sciences (head Prof. RNDr. J. Zvárová, DrSc).

¹³ Groups of attributes, i.e. *physical examination* are defined in the STULONG data set.

¹⁴ The analytical questions can be found at <http://euromise.vse.cz/challenge2004/tasks.html>

- (d) *BMI*
 - (e) *Blood pressure*
 - (f) *Level of total cholesterol, HDL cholesterol, triglycerides*
2. *What are the relations between physical activity at work and in free time and the following characteristics of men in the respective groups:*
 - (a) *Smoking*
 - (b) *Alcohol consumption*
 - (c) *BMI*
 - (d) *Blood pressure*
 - (e) *Level of total cholesterol, HDL cholesterol, triglycerides*
 3. *What are the relations between alcohol consumption and the following characteristics of men in the respective groups:*
 - (a) *Smoking*
 - (b) *BMI*
 - (c) *Blood pressure*
 - (d) *Level of total cholesterol, HDL cholesterol, triglycerides*

The experiment consisted of two steps. The first tried to answer the questions without usage of disjunctions. The second step used the task settings from the first step and allowed disjunctions of length 2. We applied the *double founded implication* quantifier with settings $p=0.9$, $Base=0.1$ and the *founded equivalence* quantifier with settings $p=0.9$ and $Base=0.1$. Below stated are the goals of the experiment:

1. Show the difference between using and not using disjunctions.
2. Use disjunctions with *strict quantifiers*.
3. Find interesting rules that contain disjunction.

4.4 Performance

It is shown in [14] that the 4FT procedure operation time without disjunctions is approximately linear to number of rows of the data matrix. Moreover, the bit string approach takes advantage of fast instructions in the processor, which makes running times acceptable for most tasks. When using disjunctions, the operation time rises due to the fact, that search space is not reduced by adding conjunctions decreasing *support* of the rule. However, practical experience shows that there is no need to be concerned, because the running times are acceptable.

In our experiment we used a Pentium M 1,7GHz processor with 1 GB of RAM and Windows XP. We noted the running times of tasks designed to answer proposed analytical questions. Without disjunctions, minimal running time was 0.310 seconds, maximal running time was 2.543 seconds and average running time was 0.829 seconds¹⁵. When using disjunctions, minimal running time was 0.370 seconds, maximal running time was 345.997 and average running time was 38.9 seconds. In the maximum case, procedure constructed and verified almost 8 million contingency tables.

¹⁵ Performance tests between the 4FT procedure implemented in Ferda and LISp-Miner can be found in [10].

4.5 Results

After conducting the first step of the experiment, we found 0 rules for 14 of 15 analytical questions and one rule for question 3.(b). This result confirmed our presumption, that *double founded implication* and *founded equivalence* are *strict quantifiers*. When using disjunctions, we found minimum 1 and maximum 185 rules per analytical question. Although we agree that number of rules found is not a good metrics of measuring performance of new data mining technique, we think that the shift from zero rules found to non-zero rules found is significant.

Let us consider the significance of the rules. In order to reduce the amount of rules presented, we consider for simplicity only the analytical question 3.(b). We may limit our analysis, because all the rules found show similar characteristics. Possible rules answering the question were presented as examples throughout the article. Moreover, a rule for this analytical question was found during step one of the experiment.

$$Beer7(No) \Leftrightarrow_{p=0.986, Base=0.968} BMI(Normalweight, Overweight)$$

is the rule found without disjunction usage. It can be explained by the fact that 7 degree beer is very rare and it was mainly used for hydration of manual workers in extremely hot working environment (glassmakers or metallurgists). Therefore majority of population did not drink this type of beer.

Table 2. Rules found

Antecedent	Succedent	DFI	FE	Base
Beer10(No) \vee Beer12(No)	BMI(Normal weight, Overweight)	0.929	0.932	0.931
Beer12(No) \vee Wine(Yes)	BMI(Normal weight, Overweight)	0.906	0.909	0.909
Beer12(No) \vee Liquor(Yes)	BMI(Normal weight, Overweight)	0.905	0.908	0.909
Beer12(No) \vee Alcohol(Occasionally)	BMI(Normal weight, Overweight)	0.902	0.904	0.904
Beer12(No) \vee BeerDaily(<1 liter)	BMI(Normal weight, Overweight)	0.946	0.948	0.948
Beer12(No) \vee WineDaily(< $\frac{1}{2}$ liter)	BMI(Normal weight, Overweight)	0.9	0.903	0.903
Wine(No) \vee WineDaily(< $\frac{1}{2}$ liter)	BMI(Normal weight, Overweight)	0.951	0.951	0.951
Liquor(No) \vee LiquorDaily(<1 dL)	BMI(Normal weight, Overweight)	0.923	0.924	0.924

Table 2 shows rules found with disjunction usage. The rules were valid both for *double founded implication* (DFI) and *founded equivalence*(FE) quantifiers. Numbers in these columns represent actual values of quantifiers, $\frac{a}{a+b+c}$ for *double founded implication*, $\frac{a+d}{a+b+c+d}$ for *founded equivalence* and $\frac{a}{a+b+c+d}$ for *Base*. According to high values of both quantifiers and also to the high support of the rules (the *Base* parameter), we have found very strong rules containing

disjunctions. Despite the initial concerns about interpretability of rules with disjunctions, the rules can be easily interpreted and comprehended. We are aware of the fact, that the rules do not show any surprising relations¹⁶. Yet they show strong relations in data, where almost no relations without disjunction usage was discovered.

To conclude the experiment, all three goals from section 4.3 were reached. We showed the difference of mining with and without disjunctions by getting almost none rules without disjunctions and more rules with disjunctions. We managed to use *strict quantifiers* and we also found interesting rules containing disjunctions. The experiment also raised many questions about further development, some of which will be discussed in the following section 5.

5 Further Research

There are several directions to improve disjunctions using in association mining. This section explains the directions in more detail. The first direction is optimization of disjunctions verifications. We showed in section 4.4 that average running times for average size tasks is acceptable. However there is still room for optimizing. As was stated before, one cannot apply pruning used with conjunctions. Solution can lie in ordering of *basic Boolean attributes* according to their support. However the ordering itself can be a significant performance problem.

Another direction is to implement generation of prime rules only. The theory about prime rules is explained in 5, 12 includes information about deduction rules, which can be used when checking prime property of an association rules.

When evaluating results of our experiments, we came across an interesting coincidence between values of quantifiers, mainly the *Base* and *p* values of *double founded implication* and *founded equivalence* quantifiers. A natural question occurs whether this is only a coincidence or if it is some kind of functional dependence. The quantifiers are known for years, yet without disjunctions we were not able on any data to mine a reasonable amount of rules to show the coincidence¹⁷. Examination of quantifiers as functions $f : \mathfrak{R}^4 \rightarrow \mathfrak{R}$ by finding functional extremes with the aid of calculus is another direction of further research, which would answer the question of coincidence or dependence between quantifiers.

Boolean attribute was presented in the article. The attribute can reach values *false* or *true* (0 or 1). *Fuzzy attribute* can also be defined, reaching values from the interval $(0, 1)$. Then fuzzy *4ft tables* can be constructed and fuzzy quantifiers can be defined¹⁸. 7 is the inspiration for this direction.

Last, but not least direction of further research is cooperation with domain experts to evaluate usability of rules with disjunctions. We mined over medical data and presented association rules comprehensible to non medical experts. Presenting the rules found to the medical experts provides valuable feedback for us.

¹⁶ This may be a problem of mining association rules in general.

¹⁷ This corresponds to considering the quantifiers as *strict*.

¹⁸ Naturally, some of the existing quantifiers could not be used.

6 Conclusion

We present an enhancement of association mining with the possibility of setting disjunctions between the items. The classical *a priori* algorithm was not suitable for disjunctions. Instead older GUHA method was applied. The 4FT procedure is used, which mines for rules in form $\varphi \approx \psi$ where φ and ψ are *Boolean attributes* and \approx is a *4ft-quantifier*. *Boolean attribute* is a recursive structure, where disjunction can be used. 4FT procedure was implemented in our Ferda data mining tool.

Experiments were conducted to show the usability of disjunctions in association mining. We tried to answer number of analytical questions from the STU-LONG medical data set containing statistical information of atherosclerosis risk factors. We applied *double founded implication* and *founded equivalence* quantifiers, which are considered to be *strict*, that is to return no rules in most cases. The experiments showed difference between mining with and without disjunctions and found strong interpretable rules containing disjunctions in the data. The experiments also showed some issues, which should be subjects of further research.

Acknowledgements

This work was supported by the project MSM6138439910 of the Ministry of Education of the Czech Republic, project IG407056 of University of Economics, Prague and by the project 201/05/0325 of the Czech Science Foundation. We acknowledge the contribution of our research colleagues Jan Rauch and Daniel Kupka for valuable comments and reviews.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proc. of the ACM SIGMOD Conference on Management of Data, pp. 207–216
2. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.: Fast discovery of association rules. In: Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.) *Advances in Knowledge Discovery and Data Mining*, pp. 307–328. AAAI Press, Menlo Park (1996)
3. Hájek, P.: The GUHA Method in the Last Century and Now. *Znalosti*. In: *Conference on Data Mining, Brno 2006*, pp. 10–20 (in Czech) (2006)
4. Hájek, P., Havel, I., Chytil, M.: The GUHA method of automatic hypotheses determination. *Computing* 1, 293–308 (1966)
5. Hájek, P., Havránek, T.: *Mechanising Hypothesis Formation - Mathematical Foundations for a General Theory*. Springer, Heidelberg (1978)
6. Hájek, P., Holeňa, M.: Formal logics of discovery and hypothesis formation by machine. *Theoretical Computer Science* 292, 345–357 (2003)
7. Holeňa, M.: Fuzzy hypotheses testing in framework of fuzzy logic. *Fuzzy Sets and Systems*, 149, pp. 229–252

8. KDNuggets Polls: Data mining/analytic techniques you use frequently. www.kdnuggets.com/polls/2005/data_mining_techniques.htm
9. Kováč, M., Kuchař, T., Kuzmin, A., Ralbovský, M.: Ferda, New Visual Environment for Data Mining. Znalosti, Conference on Data Mining, Hradec Králové, pp. 118–129 (in Czech) (2006)
10. Kuchař, T.: Experimental GUHA Procedures, Master Thesis, Faculty of Mathematics and Physics, Charles University, Prague (in Czech) (2006)
11. Kupka, D.: User support 4ft-Miner procedure for Data Mining. Master Thesis, Faculty of Mathematics and Physics, Charles University, Prague (in Czech) (2006)
12. Rauch, J.: Logic of Association Rules. Applied Intelligence 22(1), 9–28
13. Rauch, J.: Some Remarks on Computer Realisations of GUHA Procedures. International Journal of Man-Machine Studies 10, 23–28 (1978)
14. Rauch, J., Šimáunek, M.: An Alternative Approach to Mining Association Rules. In: Lin, T.Y., Ohsuga, S., Liao, C.J., Tsumoto, S. (eds.) Foundations of Data Mining and Knowledge Discovery, pp. 219–239. Springer, Heidelberg (2005)
15. Rauch, J., Šimáunek, M.: GUHA Method and Granular Computing. Proceedings of IEEE International Conference on Granular Computing (2005), <http://www.cs.sjsu.edu/~grc/grc2005/index.html>

Author Index

- Aouad, Lamine M. 120
- Barbeiro, Paulo 92
- Beringer, Jürgen 34
- Berka, Petr 135
- Bichindaritz, Isabelle 184
- Bondu, Alexis 228
- Borrajo, M. Lourdes 242
- Böttcher, Mirko 255
- Camacho, Rui 307
- Corchado, E.S. 242
- Corchado, Juan M. 242
- Costa da Silva, Josenildo 318
- de Zeeuw, Paul M. 296
- Djeraba, Chabane 107
- Dourado, Antonio 92
- Englund, Cristofer 214
- Faez, Karim 63
- Farkas, Richárd 163
- Feldman, Ronen 283
- Ferreira, Edgar 92
- Grossman, Robert 77
- Gupta, Chetan 77
- Hanias, M.P. 329
- Hüllermeier, Eyke 34
- Kanan, Hamidreza Rashidy 63
- Karras, D.A. 329
- Kechadi, Tahar M. 120
- Klusck, Matthias 318
- Kuchař, Tomáš 339
- Kunegis, Jérôme 269
- Kuri-Morales, Angel 199
- Labský, Martin 135
- Lakshmi, K. 148
- Le-Khac, Nhien-An 120
- Lemaire, Vincent 228
- Michael, Shaul Ben 283
- Mukherjee, Saswati 148
- Nauck, Detlef 255
- Nouretdinov, Ilia 15
- Ormándi, Róbert 163
- Orsenigo, Carlotta 49
- Pauwels, Eric J. 296
- Pellicer, M.A. 242
- Perner, Petra 21
- Poulain, Barbara 228
- Ralbovský, Martin 339
- Ramos, Ruy 307
- Ranguelova, Elena 296
- Richter, Michael M. 1
- Rodríguez-Erazo, Fátima 199
- Rumyantsev, Alexander 173
- Schmidt, Rainer 173
- Schmidt, Stephan 269
- Simovici, Dan A. 107
- Spott, Martin 255
- Szarvas, György 163
- Taheri, Sayyed Mostafa 63
- Urruty, Thierry 107
- Vercellis, Carlo 49
- Verikas, Antanas 214
- Vorobieva, Olga 173