

Petra Perner
Atsushi Imiya (Eds.)

LNAI 3587

Machine Learning and Data Mining in Pattern Recognition

4th International Conference, MLDM 2005
Leipzig, Germany, July 2005
Proceedings

 Springer

Lecture Notes in Artificial Intelligence 3587

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Petra Perner Atsushi Imiya (Eds.)

Machine Learning and Data Mining in Pattern Recognition

4th International Conference, MLDM 2005
Leipzig, Germany, July 9-11, 2005
Proceedings

Series Editors

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

Volume Editors

Petra Perner

Institute of Computer Vision and Applied Computer Sciences, IBAI
Körnerstr 10, 04107 Leipzig, Germany
E-mail: ibaipermer@aol.com

Atsushi Imiya

Chiba University, Department of Information and Image Sciences
1-33, Yayoi-cho, Inage-ku, Chiba-shi, Chiba, 263-8522, Japan
E-mail: imiya@ics.tj.chiba-u.ac.jp

Library of Congress Control Number: 2005928444

CR Subject Classification (1998): I.2, I.5, I.4, F.4.1, H.3

ISSN 0302-9743
ISBN-10 3-540-26923-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-26923-6 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

springeronline.com

© Springer-Verlag Berlin Heidelberg 2005
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11510888 06/3142 5 4 3 2 1 0

Preface

We met again in front of the statue of Gottfried Wilhelm von Leibniz in the city of Leipzig. Leibniz, a famous son of Leipzig, planned automatic logical inference using symbolic computation, aimed to collate all human knowledge. Today, artificial intelligence deals with large amounts of data and knowledge and finds new information using machine learning and data mining. Machine learning and data mining are irreplaceable subjects and tools for the theory of pattern recognition and in applications of pattern recognition such as bioinformatics and data retrieval.

This was the fourth edition of MLDM in Pattern Recognition which is the main event of Technical Committee 17 of the International Association for Pattern Recognition; it started out as a workshop and continued as a conference in 2003. Today, there are many international meetings which are titled “machine learning” and “data mining”, whose topics are text mining, knowledge discovery, and applications. This meeting from the first focused on aspects of machine learning and data mining in pattern recognition problems. We planned to reorganize classical and well-established pattern recognition paradigms from the viewpoints of machine learning and data mining. Though it was a challenging program in the late 1990s, the idea has inspired new starting points in pattern recognition and effects in other areas such as cognitive computer vision.

For this edition the Program Committee received 103 submissions from 20 countries. After the peer-review process, we accepted 58 papers for presentation. We deeply thank the members of the Program Committee and the reviewers, who examined some difficult papers from broad areas and applications. We also thank the members of the Institute of Applied Computer Sciences, Leipzig, Germany who ran the conference 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.

This proceedings also includes a special selection of papers from the Industrial Conference on Data Mining, ICDM-Leipzig 2005, which we think are also interesting for the audience of this book. We also thank the members of the Program Committee of ICDM 2005 for their valuable work, and all the speakers who made this event a success.

Leipzig,
July 2005

Petra Perner
Atsushi Imiya

Machine Learning and Data Mining in Pattern Recognition MLDM 2005

Co-chairs

Petra Perner
Institute of Computer Vision and Applied
Computer Sciences, IBAI, Leipzig, Germany

Atsushi Imiya
Chiba University, Japan

Program Committee

Agnar Aamodt
NTNU, Norway

Horst Bunke
University of Bern, Switzerland

Max Bramer
University of Portsmouth, UK

Krzysztof Cios
University of Colorado, USA

John Debenham
University of Technology, Australia

Dragan Gamberger
Rudjer Boskovic Institute, Croatia

Lothar Gierl
University of Rostock, Germany

Howard J. Hamilton
University of Regina, Canada

Thomas S. Huang
University of Illinois, USA

Atsushi Imiya
Chiba University, Japan

Horace Ip
City University, Hong Kong, China

Herbert Jahn
Aero Space Center, Germany

Longin Jan Latecki
Temple University, Philadelphia, USA

Adam Krzyzak
Concordia University, Montreal, Canada

Brian Lovell
University of Queensland, Australia

Ryszard Michalski
George Mason University, USA

Donato Malerba
University of Bari, Italy

Fabio Roli
University of Cagliari, Italy

Gabriella Sanniti di Baja
Istituto di Cibernetica, Italy

Michele Sebag
Ecole Polytechnique, France

Arnold Smeulders
University of Amsterdam, The Netherlands

Patrick Wang
Northeastern University, USA

Harry Wechsler
George Mason University, USA

Sholom Weiss
IBM Yorktown Heights, USA

Additional Reviewers

Jörg Cassens
Diego Federici
Magnus Lie Hetland
Helge Langseth
Pavel Petrovic
Amund Tveit
Tomasz Rybak
Marek Kretowski
Zenon A. Sosnowski
Adam Schenker
Bertrand Le Saux
Christophe Irniger
Andreas Schlapbach
Rainer Schmidt
Jilin Tu
Hanning Zhou
Huazhong Ning
Ming Liu
George Q. Chui
Guoqin Cui
Dennis Lin
Charlie Dagli
Zhenqiu Zhang
Yisong Chen
Hau San Wong
Shuo Li
Lars Kulik
Rolf Lakaemper
Aleksandar Lazarevic

Slobodan Vucetic
Caruso Costantina
Annalisa Appice
Antonio Varlaro
Corrado Loglisci
Michelangelo Ceci
Margherita Berardi
Gian Luca Marcialis
Giorgio Giacinto
Roberto Perdisci
Giorgio Fumera
Luca Didaci
Alessandra Serrau
Olivier Teytaud
Gelly Sylvain
Mary Jeremie
Nicolas Bredeche
Samuel Landau
Antoine Cornuejols
Jerome Aze
Mathieu Roche
Thang Viet Pham
Gertjan J. Burghouts
Lin Mei
Matthew Ma
Saharon Rosset
Tong Zhang
Ricardo Vilalta

Industrial Conference on Data Mining ICDM 2005

Co-chairs

Petra Perner	Institute of Computer Vision and Applied Computer Sciences, IBAI, Leipzig, Germany
Andrea Ahlemeyer-Stubbe	ECDM, Gengenbach, Germany

Program Committee

Klaus-Dieter Althoff	Fraunhofer IESE, Germany
Chid Apte	IBM Yorktown Heights, USA
Isabelle Bichindaritz	University of Washington, USA
Leon Bobrowski	Bialystok Technical University, Poland
Marc Boulle	France Télécom, France
Da Deng	University of Otago, New Zealand
Ron S. Kenett	KPA Ltd., Raanana, Israel
Eric Pauwels	CWI, The Netherlands
Rainer Schmidt	University of Rostock, Germany
Stijn Viaene	KULeuven, Belgium
Rob A. Vingerhoeds	Ecole Nationale d'Ingénieurs de Tarbes, France

Table of Contents

Classification and Model Estimation

On ECOC as Binary Ensemble Classifiers <i>J. Ko, E. Kim</i>	1
Incremental Classification Rules Based on Association Rules Using Formal Concept Analysis <i>Anamika Gupta, Naveen Kumar, Vasudha Bhatnagar</i>	11
Parameter Inference of Cost-Sensitive Boosting Algorithms <i>Yanmin Sun, Andrew K.C. Wong, Yang Wang</i>	21
Finite Mixture Models with Negative Components <i>Baibo Zhang, Changshui Zhang</i>	31
MML-Based Approach for Finite Dirichlet Mixture Estimation and Selection <i>Nizar Bouguila, Djemel Ziou</i>	42
Principles of Multi-kernel Data Mining <i>Vadim Mottl, Olga Krasotkina, Oleg Seredin, Ilya Muchnik</i>	52

Neural Methods

Comparative Analysis of Genetic Algorithm, Simulated Annealing and Cutting Angle Method for Artificial Neural Networks <i>Ranadhir Ghosh, Moumita Ghosh, John Yearwood, Adil Bagirov</i>	62
Determining Regularization Parameters for Derivative Free Neural Learning <i>Ranadhir Ghosh, Moumita Ghosh, John Yearwood, Adil Bagirov</i>	71
A Comprehensible SOM-Based Scoring System <i>Johan Huysmans, Bart Baesens, Jan Vanthienen</i>	80

Subspace Methods

The Convex Subclass Method: Combinatorial Classifier Based on a Family of Convex Sets <i>Ichigaku Takigawa, Mineichi Kudo, Atsuyoshi Nakamura</i>	90
---	----

SSC: Statistical Subspace Clustering
Laurent Candillier, Isabella Tellier, Fabien Torre,
Olivier Bousquet 100

Understanding Patterns with Different Subspace Classification
Gero Szepannek, Karsten Luebke, Claus Weihs 110

Clustering: Basics

Using Clustering to Learn Distance Functions for Supervised Similarity
 Assessment
Christoph F. Eick, Alain Rouhana, Abraham Bagherjeiran,
Ricardo Vilalta 120

Linear Manifold Clustering
Robert Haralick, Rave Harpaz 132

Universal Clustering with Regularization in Probabilistic Space
Vladimir Nikulin, Alex J. Smola 142

Acquisition of Concept Descriptions by Conceptual Clustering
Silke Jänichen, Petra Perner 153

Applications of Clustering

Clustering Large Dynamic Datasets Using Exemplar Points
William Sia, Mihai M. Lazarescu 163

Birds of a Feather Surf Together: Using Clustering Methods to Improve
 Navigation Prediction from Internet Log Files
Martin Halvey, Mark T. Keane, Barry Smyth 174

Alarm Clustering for Intrusion Detection Systems in Computer
 Networks
Giorgio Giacinto, Roberto Perdisci, Fabio Roli 184

Clustering Document Images Using Graph Summaries
Eugen Barbu, Pierre Héroux, Sébastien Adam, Eric Trupin 194

Feature Grouping, Discretization, Selection and Transformation

Feature Selection Method Using Preferences Aggregation
Gaëlle Legrand, Nicolas Nicoloyannis 203

Ranked Modelling with Feature Selection Based on the <i>CPL</i> Criterion Functions <i>Leon Bobrowski</i>	218
A Grouping Method for Categorical Attributes Having Very Large Number of Values <i>Marc Boullé</i>	228
Unsupervised Learning of Visual Feature Hierarchies <i>Fabien Scalzo, Justus Piater</i>	243
Multivariate Discretization by Recursive Supervised Bipartition of Graph <i>Sylvain Ferrandiz, Marc Boullé</i>	253
CorePhrase: Keyphrase Extraction for Document Clustering <i>Khaled M. Hammouda, Diego N. Matute, Mohamed S. Kamel</i>	265
A New Multidimensional Feature Transformation for Linear Classifiers and Its Applications <i>EunSang Bak</i>	275
Applications in Medicine	
Comparison of FLDA, MLP and SVM in Diagnosis of Lung Nodule <i>Aristófanés Corrêa Silva, Anselmo Cardoso de Paiva, Alexandre Cesar Muniz de Oliveira</i>	285
Diagnosis of Lung Nodule Using Reinforcement Learning and Geometric Measures <i>Aristófanés Corrêa Silva, Valdeci Ribeiro da Silva Junior, Areolino de Almeida Neto, Anselmo Cardoso de Paiva</i>	295
Iris Recognition Algorithm Based on Point Covering of High-Dimensional Space and Neural Network <i>Wenming Cao, Jianhui Hu, Gang Xiao, Shoujue Wang</i>	305
Automatic Clinical Image Segmentation Using Pathological Modelling, PCA and SVM <i>Shuo Li, Thomas Fevens, Adam Krzyżak, Song Li</i>	314
Improved MRI Mining by Integrating Support Vector Machine Priors in the Bayesian Restoration <i>D.A. Karras, B.G. Mertzios, D. Graveron-Demilly, D. van Ormondt</i>	325

Prediction of Secondary Protein Structure Content from Primary Sequence Alone – A Feature Selection Based Approach
Lukasz Kurgan, Leila Homaeian 334

Alternative Clustering by Utilizing Multi-objective Genetic Algorithm with Linked-List Based Chromosome Encoding
Jun Du, Emin Erkan Korkmaz, Reda Alhaji, Ken Barker 346

Time Series and Sequential Pattern Mining

Embedding Time Series Data for Classification
Akira Hayashi, Yuko Mizuhara, Nobuo Suematsu 356

Analysis of Time Series of Graphs: Prediction of Node Presence by Means of Decision Tree Learning
Horst Bunke, Peter Dickinson, Christophe Irrniger, Miro Kraetzl 366

Disjunctive Sequential Patterns on Single Data Sequence and Its Anti-monotonicity
Kazuhiro Shimizu, Takao Miura 376

Mining Expressive Temporal Associations from Complex Data
Keith A. Pray, Carolina Ruiz 384

Statistical Supports for Frequent Itemsets on Data Streams
Pierre-Alain Laur, Jean-Emile Symphor, Richard Nock, Pascal Poncelet 395

Mining Images in Computer Vision

Autonomous Vehicle Steering Based on Evaluative Feedback by Reinforcement Learning
Klaus-Dieter Kuhnert, Michael Krödel 405

Cost Integration in Multi-step Viewpoint Selection for Object Recognition
Christian Derichs, Frank Deinzer, Heinrich Niemann 415

Support Vector Machine Experiments for Road Recognition in High Resolution Images
J.Y. Lai, A. Sowmya, J. Trinder 426

An Automatic Face Recognition System in the Near Infrared Spectrum <i>Shuyuan Zhao, Rolf-Rainer Grigat</i>	437
---	-----

Mining Images and Texture

Hierarchical Partitions for Content Image Retrieval from Large-Scale Database <i>Dmitry Kinoshenko, Vladimir Mashtalir, Elena Yegorova, Vladimir Vinarsky</i>	445
Optimising the Choice of Colours of an Image Database for Dichromats <i>Vassili Kovalev, Maria Petrou</i>	456
An Approach to Mining Picture Objects Based on Textual Cues <i>Adeoye I. Adegorite, Otman A. Basir, Mohamed S. Kamel, Khaled B. Shaban</i>	466

Mining Motion from Sequence

Activity and Motion Detection Based on Measuring Texture Change <i>Longin Jan Latecki, Roland Miezancko, Dragoljub Pokrajac</i>	476
A New Approach to Human Motion Sequence Recognition with Application to Diving Actions <i>Shiming Xiang, Changshui Zhang, Xiaoping Chen, Naijiang Lu</i>	487
Dominant Plane Detection Using Optical Flow and Independent Component Analysis <i>Naoya Ohnishi, Atsushi Imiya</i>	497

Speech Analysis

Neural Expert Model Applied to Phonemes Recognition <i>Halima Bahi, Mokhtar Sellami</i>	507
An Evidential Reasoning Approach to Weighted Combination of Classifiers for Word Sense Disambiguation <i>Cuong Anh Le, Van-Nam Huynh, Akira Shimazu</i>	516

Aspects of Data Mining

Signature-Based Approach for Intrusion Detection <i>Bon K. Sy</i>	526
--	-----

Discovery of Hidden Correlations in a Local Transaction Database
Based on Differences of Correlations
Tsuyoshi Taniguchi, Makoto Haraguchi, Yoshiaki Okubo 537

An Integrated Approach for Mining Meta-rules
*Feiyue Ye, Jiandong Wang, Shiliang Wu, Huiping Chen,
Tianqiang Huang, Li Tao* 549

Data Mining on Crash Simulation Data
*Annette Kuhlmann, Ralf-Michael Vetter, Christoph Lübbing,
Clemens-August Thole* 558

Text Mining

Pattern Mining Across Domain-Specific Text Collections
Lee Gillam, Khurshid Ahmad 570

Text Classification Using Small Number of Features
Masoud Makrehchi, Mohamed S. Kamel 580

Low-Level Cursive Word Representation Based on Geometric
Decomposition
*Jian-xiong Dong, Adam Krzyżak, Ching Y. Suen,
Dominique Ponson* 590

Special Track: Industrial Applications of Data Mining

Supervised Evaluation of Dataset Partitions: Advantages and Practice
Sylvain Ferrandiz, Marc Boullé 600

Inference on Distributed Data Clustering
Josenildo C. da Silva, Matthias Klusch 610

A Novel Approach of Multilevel Positive and Negative Association
Rule Mining for Spatial Databases
L.K. Sharma, O.P. Vyas, U.S. Tiwary, R. Vyas 620

Mixture Random Effect Model Based Meta-analysis for Medical Data
Mining
Yinglong Xia, Shifeng Weng, Changshui Zhang, Shao Li 630

Semantic Analysis of Association Rules via Item Response Theory
Shinichi Hamano, Masako Sato 641

Temporal Approach to Association Rule Mining Using T-Tree and P-Tree <i>Keshri Verma, O.P. Vyas, Ranjana Vyas</i>	651
Aquaculture Feature Extraction from Satellite Image Using Independent Component Analysis <i>JongGyu Han, KwangHoon Chi, YeonKwang Yeon</i>	660
Modeling the Organoleptic Properties of Matured Wine Distillates <i>S.B. Kotsiantis, G.E. Tsekouras, C. Raptis, P.E. Pintelas</i>	667
Bagging Random Trees for Estimation of Tissue Softness <i>S.B. Kotsiantis, G.E. Tsekouras, P.E. Pintelas</i>	674
Concept Mining for Indexing Medical Literature <i>Isabelle Bichindaritz, Sarada Akkineni</i>	682
Author Index	693

On ECOC as Binary Ensemble Classifiers

J. Ko¹ and E. Kim²

¹ Dept. of Computer Engineering, Kumoh National Institute of Technology,
1, Yangho-dong, Gumi, Gyeongbuk 730-701, Korea
nonezero@kumloh.ac.kr,

² National Computerization Agency,
NCA Bldg 77, Mugyo-dong, Jung-gu, Seoul 100-775, Korea
outframe@nca.go.kr

Abstract. The Error-Correcting Output Codes (ECOC) is a representative approach of the binary ensemble classifiers for solving multi-class problems. There have been so many researches on an output coding method built on an ECOC foundation. In this paper, we revisit representative conventional ECOC methods in an overlapped learning viewpoint. For this purpose, we propose new OPC based output coding methods in the ECOC point of view, and define a new measure to describe their properties. From the experiment on a face recognition domain, we investigate whether a problem complexity is more important than the overlapped learning or an error correction concept.

1 Introduction

The Error-Correcting Output Codes (ECOC) [1] is one of the binary ensemble classifiers for solving multi-class problems. The ECOC has been dominant theoretical foundation in output coding methods [2-6] that decompose a complex multi-class problem into a set of binary problems and then reconstructs the outputs of binary classifiers for each binary problem. The performance of output coding methods depends on base binary classifiers. It needs to revisit the ECOC concept, since the Support Vector Machines (SVM) [7] that can produce a complex nonlinear decision boundary with a good generalization performance is available as a base classifier for output coding methods.

The ECOC has two principals with respect to a codes design in which the codes concern both how to decompose a multi-class problem into several binary ones and how to decide a final decision. One principal is to enlarge the minimum hamming distance of a decomposition matrix. The other is to enlarge the row separability to increase the diversity among binary problems. A high diversity reduces an error-correlation among binary machines [8]. By enlarging the length of codewords [9], we can easily increase the hamming distance of the decomposition matrix at the cost of generating a large number of binary problems. In this circumstance, each class can be learned redundantly in several binary machines, we call it *overlapped learning*. By increasing the error-correction ability through the overlapped learning, we have been

able to improve performance of a conventional ECOC with a hamming decoding. The hamming decoding closely concerns the hamming distance of the decomposition matrix.

In a *generalized ECOC* [9] that includes 0 elements as well as -1 and $+1$ in the decomposition matrix, i.e., it has a triple codes (on the other side, a *conventional ECOC* consists of -1 and $+1$, i.e., it has a binary codes), we cannot directly compute the hamming distance. A new distance, a generalized hamming distance, is defined by [9], where the distance between the 0 element and the others is 0.5. The primary motivation of the conventional ECOC has been the overlapped learning of classes built on binary codes. The generalized ECOC does not insist on the binary codes any more, and the SVM used for a binary classifier can produce a real-valued confidence output that can be useful information for discriminating classes.

In this paper, we revisit ECOC with respect to the generalized ECOC by comparing and empirically analyzing certain properties of the representative ECOC methods, such as One-Per-Class (OPC) [11], All-Pairs [12], Correcting Classifier (CC) [10] and our proposed OPC-based methods designed on conventional ECOC concept. Further, we give an empirical conclusion on a codes design, which is limited to our experiment on face recognition.

2 One-per-Class Variants with ECOC Concept

In this section, we firstly formulate the *output coding* method (a generalized ECOC) in two steps: decomposition and reconstruction. Then, we propose new OPC based output coding method with ECOC concept, and define a new measure to describe their properties. Further, we describe later the performance of ECOC *with margin decoding*, which uses the real-valued output of a machine, using a newly defined problem complexity measure in the experiment. The OPC with hamming decoding has no error correction ability, so we begin by introducing additional machines to endow it with an error correcting ability.

2.1 Decomposition and Decoding

Decomposition (Encoding): A decomposition matrix, $D \in \{-1, 0, +1\}^{L \times K}$, specifies K classes to train L machines (dichotomizers), f_1, \dots, f_L . The machines f_l is trained according to the row $D(l, \cdot)$. If $D(l, k) = +1$, all examples of class k are positive and if $D(l, k) = -1$, all examples of class k are negative, and if $D(l, k) = 0$ none of the examples of class k participates in the training of f_l . The column of D is called *code-words*. The entry “0” is introduced by [9]. Hence, some examples for $D(l, k) = 0$ can be omitted in the training phase.

We can formulate two separated super classes C_l^+ and C_l^- for the machine f_l as follows: $C_l^+ = \{C_k \mid D(l, k) = 1\}$, $C_l^- = \{C_k \mid D(l, k) = -1\}$.

Decoding (Reconstruction): In the decoding step, a simple nearest-neighbor rule is commonly used. The class output is selected that maximizes some similarity measure $s: \mathbf{R}^L \times \{-1, 0, 1\}^L \rightarrow [-\infty, \infty]$, between $f(\mathbf{x})$ and column $D(\cdot, k)$.

$$class_output = \arg \max_k s(f(\mathbf{x}), D(\cdot, k)) \quad (1)$$

We call it a *margin decoding*, equation (2), a similarity measure based on a *margin*, defined as $y \cdot f(x)$ [9]. When classifier outputs a hard decision, $h(\mathbf{x}) \in \{-1, 1\}$, the method is called *hamming decoding*, equation (3).

$$s(f(\mathbf{x}), D(\cdot, k)) = \sum_l f_l(\mathbf{x}) D(l, k) \quad (2)$$

$$s_H(h(\mathbf{x}), D(\cdot, k)) = 0.5 \times \sum_l (1 + h_l(\mathbf{x})) D(l, k) \quad (3)$$

2.2 New Decompositions

Tree-Based Decomposition: We design the tree structure for getting additional machines as well as those of generated by OPC. We adopt binary tree and distribute the classes of a parent node to its child nodes in a crossing manner. By the crossing manner, we can achieve the diversity of the binary problems with our proposed decomposing method as follows. Each node except for the root node makes one row in a decomposition matrix by assigning a positive value for classes that the node has, and a negative value for the other classes in the sibling nodes. The root node gives a positive value for the half of the whole classes and a negative value for the remainder. Fig. 1 shows a generated decomposition tree and a decomposition matrix on 8 classes.

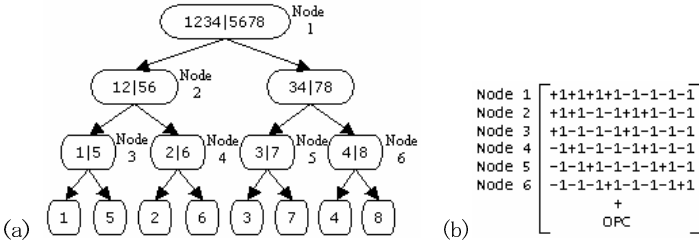


Fig. 1. Decomposition matrix of Tree-based scheme for 8 classes. (a) tree-structure on 8 classes. (b) Its decomposition matrix

When the number of classes is K , the $2 \times (K - 1)$ problems are generated. The difference between the number of classes being a positive class and the number of classes being a negative class varies according to the level of depth of the tree, so each binary problem can have the different level of complexity. Therefore, it is desirable to introduce weights into the decoding process to handle a different complexity among problems.

N-Shift Decomposition: In this scheme, we first decide the number of positive classes N , and then form the first row of a decomposition matrix by setting N elements from left as positive ones and the remainder as negative ones. The rest rows are easily constructed by right-shifting the elements of the preceding row. Finally, OPC decomposition matrix is added to it. When the number of classes is K , the $2 \times K$

problems are generated. Fig. 2 shows two examples of a generated decomposition matrix having different N values, 2 and 3, respectively, when K is 4.

$$\begin{array}{cc} \begin{bmatrix} +1 & +1 & -1 & -1 \\ -1 & +1 & +1 & -1 \\ -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 \\ + \\ \text{OPC} \end{bmatrix} & \begin{bmatrix} +1 & +1 & +1 & -1 \\ -1 & +1 & +1 & +1 \\ +1 & -1 & +1 & +1 \\ +1 & +1 & -1 & +1 \\ + \\ \text{OPC} \end{bmatrix} \\ N = 2 & N = 3 \end{array}$$

Fig. 2. Decomposition matrix of 2-Shift and 3-Shift for 4 classes

2.3 New Decodings

It is undesirable to deal with the outputs of the machines equally where each machine is trained with a problem having different level of complexity. There are two possible solutions to this problem: One is to utilize the different level of output for class decision, and the other is to adopt a weighed output. In this section, we propose the *relative distance decoding* for the former, and the *weighted decoding* for the latter respective.

Relative Distance Decoding: The machine has different scale outputs for two classes, so the same outputs should be understood differently. As an example, consider that, for samples belonging to class i , the machine habitually generates 0.8 , and for samples belonging to class j , 0.5 . The habit of generating uneven outputs for classes is formed during the learning process, and can be used for discriminating classes. To utilize this information, we introduce an *average template*. The average template is constructed by calculating the average of output for each machine as follows:

$$D'(i, j) = \left(\sum_{\mathbf{x} \in C_j} f_i(\mathbf{x}) \right) / |C_j| \quad (4)$$

where $|C_j|$ means the number of samples belonging to the class j . The following equation calculates the similarity between a given input and a considered class by the relative distance.

$$\begin{aligned} rd(f(x), D'(\cdot, k)) &= 1 / (1 + \exp(Ad + B)) \\ d &= \sum_l \|f_l(\mathbf{x}) - D'(l, k)\|_2 \end{aligned} \quad (5)$$

Both A and B constants of the exponential function and they can be usually fixed by experiment.

Weighted Decoding: As the number of positive classes increases, the complexity of the binary problem increases accordingly. There is a difference between the confidences on the outputs of a machine trained with problems having different level of complexity. To handle this problem, we introduce weighting into the decoding process. The weight for learner l , w_l , is calculated as follows:

$$w_l = 1 / \sum_k L(D(l, k))$$

$$L(D(l, k)) = \begin{cases} 1 & \text{if } D(l, k) > 0 \\ 0 & \text{else} \end{cases} \quad (6)$$

where, $L(D(l, k))$ is a function for discerning positive classes from negative classes. Then, the weighted decoding is as follows:

$$s(f(x), D(\cdot, k)) = \sum_l w_l f_l(x) D(l, k) \quad (7)$$

This decoding can be used for determining the complexity of a problem. If we adopt this measure and obtain improvement in decomposition, then we can think that the decomposition generates complex problems.

3 Intuitive Problem Complexity

We define a new measure for estimating the complexity of a machine as well as the weighted decoding. We need some measure that estimates the complexity when in designing the decomposition matrix, not in the experiment as the weighted decoding.

The magnitude of a super class, equation (2), for training a binary classifier, means that how many classes are grouped into one. Intuitively, one expects that, as the number of classes that is grouped into one increases, i.e., the magnitude of $|C_l^+|$ or $|C_l^-|$ increases, the complexity of the binary problem associated with them will increase. From this viewpoint, we can say that the most complex case is $|C_l^+| = |C_l^-| = (K/2) \gg 2$, and the easiest case is $|C_l^+| \text{ or } |C_l^-| = 1$ when the number of classes is K . In other words, if we define intuitively the problem complexity as the magnitude of the super class of a binary problem, this can be in proportion to $|C_l^+|$ and $|C_l^-|$. Let us define Intuitive Problem Complexity (IPC) as follows:

$$IPC \equiv \text{Min}(|C^+|, |C^-|) \quad (8)$$

We summarize the magnitude of each super class of different decompositions and IPC in Table 1. According to Table 1, the tree-based scheme can be considered as a very complex problem compared to other schemes. The second complex problem can be the N -Shift scheme or CC scheme up to the value of N .

Table 1. Comparison of the magnitudes of super classes and IPC

Decomposition Scheme	OPC	All-Pairs	CC	Tree-based	N -Shift
$ C^+ $	1	1	2	$K/2$	N
$ C^- $	$K-1$	1	$K-2$	$K/2$	$K-N$
IPC	1	1	2	$K/2$	N

4 Experimental Results

4.1 Data Sets

We used the ORL face dataset, which is one of the popular public datasets used in face recognition. The image set consists of 400 images, ten images for each individual. Each image for one person differs from each other in lighting, facial expression, and pose. We obtain the final training and testing dataset by applying preprocessing and Principal Component Analysis. Fig. 3 shows examples of the normalized face images produced after preprocessing.



Fig. 3. Some normalized facial images in the ORL dataset

We used all of the face images for PCA transformation, and divided them into two parts; one was used for a gallery set (reference set), and the other was used for a probe set (test set). We obtained the highest recognition rate at 48-dimension with a PCA-based rank test, which is the standard test protocol described in FERET [13]. We determined the feature dimension by employing the procedure mentioned above, because the focus of our experiments is to display the classification performance of our proposed method. To compare the properties of the output coding methods, we used the SMOBR [14], which is one of the implementations of SMO [15], with RBF kernels as a base classifier. We randomly selected five images of each person, for training and the remaining five for testing. The number of samples for training and testing is both 200 respectively and the dimension of one sample is 48. Note that the dataset has a relatively small number of samples for its high dimensional feature space. We evaluated various decoding schemes on the ORL face dataset and compared their recognition performance. Table 2 shows the decoding schemes we investigated.

Table 2. Various decoding schemes

Symbol	Meaning
HM	Hamming Decoding
MG	Margin Decoding
RD	Relative Distance Decoding
WHM	Weighted Hamming Decoding
WMG	Weighted Margin Decoding

In the subsequent section, the recognition accuracy of each decomposition scheme is presented. For those results, we calculated the recognition accuracy, varying C of SVM parameter from 1 to 10 and dispersion from 0.2 to 1.0 and chose the best recognition accuracy among them.

4.2 Properties Analysis

In this section, we compare and analyze empirically some properties of the representative output coding methods, such as OPC, All-Pairs, CC and our proposed OPC-based methods, on the following items.

Relationships Between Overlapped Learning and Hamming Decoding: The error correcting ability is related to the minimum hamming distance of a decomposition matrix, and this is obtained from the overlapped learning of classes. We investigate it empirically. The number of binary machines generated and the minimum hamming distance of each output coding method for 40 classes are summarized in Table 3. We assume that the hamming distance between zero and zero or nonzero element of a decomposition matrix is 0.5.

Table 3. Number of machines and Minimum hamming distance of decomposition schemes

Decomposition Scheme	Number of Machines		Minimum Hamming Distance	
	$K=40$	K -Class	$K=40$	K -Class
OPC	40	K	2	2
All-Pairs	780	$K(K-1)/2$	390	$(K(K-1)/2-1)/2+1$
CC	780	$K(K-1)/2$	76	$2(K-2)$
N -Shift	80	$K+K$	2	2
Tree-Based	78	$(K-2)+K$	2	2

Fig. 4 (a) presents the recognition accuracy of each decomposition scheme with hamming decoding. If we compare the recognition accuracy of Fig. 4 (a) with the number of machines and the minimum hamming distance in Table 3, we can observe that the recognition accuracy is in proportion to both the number of machines and the minimum hamming distance.

The recognition accuracy of OPC is considerably lower than those of N -Shift and Tree-based schemes in spite of their having the same hamming distances. The reason for this observation is that OPC does not retain any error correction ability because it does not conduct overlapped learning. In other words, both N -Shift and Tree-based schemes generate some extra binary machines in addition to the same machines of the OPC scheme; as a result, this allows for them to train classes in an overlapped manner, where it makes a considerable difference. Therefore, we conclude that the recognition accuracy of each decomposition scheme with hamming decoding depends on the number of machines for overlapped learning as well as its minimum hamming distance.

Hamming Decoding Versus Margin Decoding: According to Fig. 4(b), margin decoding is superior to hamming decoding for all the decomposition schemes, except for All-Pairs. This means that the margin decoding does not strongly depend on the number of machines or the minimum hamming distance. The reason for the poor accuracy of All-Pairs with margin decoding can be explained by two viewpoints as follows: First, the number of samples being used in training each machine of All-Pairs is significantly smaller than that of OPC. Secondly, the decomposition matrix includes zero elements, which means that some classes exist that are not involved in training a machine. That raises the problem of nonsense outputs. The level of the nonsense outputs problem increases as the number of classes increases.

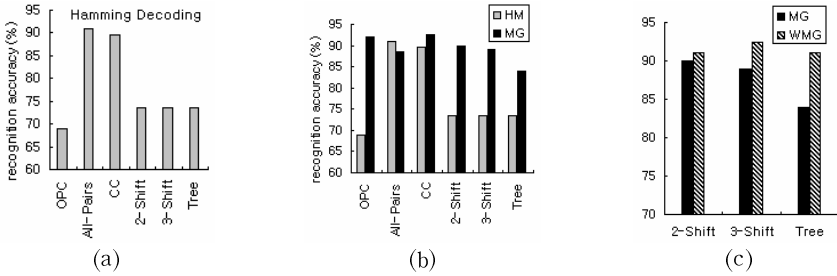


Fig. 4. Comparison of recognition accuracy (a) with hamming decoding, (b) between hamming and margin decoding, and (c) between margin and weighted margin decoding

Relationships Between Performance and Intuitive Problem Complexity: While N -Shift and Tree-Based schemes have more machines due to the overlapped learning, they are inferior to OPC in recognition accuracy. For explanation of the reason, we consider the Intuitive Problem Complexity (IPC) and the weighted decoding. The IPC of each decomposition scheme being computed using Table 1 with $K=4$, can be ordered ascendant as follow: OPC=1, 2-Shift=2, 3-Shift=3, and Tree-Based=20. This order corresponds exactly to the order of their recognition accuracy shown in Fig. 4(c).

Therefore, we infer that the overlapped learning has a strong effect when it is used with hamming decoding; however, this is not the case with margin decoding. In other words, recognition accuracy depends more on the IPC than the overlapped learning effects when we use margin decoding. Table 4 presents both the IPC and recognition accuracy on the ORL dataset.

To support this inference, We compare the recognition accuracy of N -Shift and Tree-Based schemes with margin decoding and weighted decoding respectively in Fig. 4(c). According to Table 4 and Fig. 4(c), the recognition accuracy of each decomposition scheme decreases as the IPC increases; however, their recognition accuracy is almost the same as our proposed weighted margin decoding. This means that weighted margin decoding can remove something related to the problem complexity represented by IPC. These results allow us to infer again that recognition accuracy strongly depends on the IPC of each decomposition matrix when we use margin decoding.

Table 4. Recognition Accuracy (RA) of decomposition schemes with margin decoding and IPC

Decomposition	OPC	2-Shift	3-Shift	Tree-Based
RA (%)	92.0	90.0	89.0	84.0
IPC	1	2	3	20

Performance Analysis: In Table 5, we present the recognition accuracy of the experiments on the ORL dataset with various decomposition and decoding schemes.

Table 5. Recognition accuracy (%) on the ORL face dataset

Decomposition Scheme	Decoding Scheme				
	HM	MG	RD	WHM	WMG
OPC	69.0	92.0	93.0	-	-
All-Pairs	91.0	88.5	88.5	-	-
CC	89.5	92.5	93.0	-	-
2-Shift	73.5	90.0	93.0	73.5	91.0
3-Shift	73.5	89.0	90.0	71.5	92.5
Tree-Based	73.5	84.0	85.5	75.0	91.0

When we compare the OPC and All-Pairs, with the hamming decoding, All-Pairs shows a significantly better performance than OPC, but with the margin decoding, OPC shows a better performance. Overall, OPC with the margin decoding shows slightly better performance than All-Pairs with the hamming decoding. We infer that the performance of OPC training all classes at a time is better than that of All-Pairs since the number of training face image of one person is small.

Each machine in OPC and CC trains all the classes at a time. In this case, CC shows significantly better performance than OPC in hamming decoding like All-Pairs due to its large number of machines. With margin decoding, the performance of the two machines is almost the same regardless of their differing numbers.

Consequently, when we have small number of samples, such as face images, the OPC-like schemes training all the classes at a time can be preferred, but it is unnecessary to make too many machines for the overlapped learning like the CC scheme to improve an error correcting ability at the expense of a larger IPC than OPC and All-Pairs.

5 Conclusion

In this paper, we compared and analyzed empirically certain properties of the representative output coding methods such as OPC, All-Pairs, CC and our proposed OPC-based methods with a face recognition problem. We observed the followings: Firstly, the recognition accuracy of each decomposition scheme with a hamming decoding depends on the number of machines for overlapped learning as well as its minimum hamming distance of it. Secondly, the margin decoding is superior to hamming decoding with all the decomposition schemes except for All-Pairs. The margin decoding

is slightly independent of the number of machines or the minimum hamming distance. Thirdly, we infer that an overlapped learning can have a strong effect when it is used with the hamming decoding, but this is not the case with the margin decoding. This means that recognition accuracy relies more on the IPC than the overlapped learning effects when we use the margin decoding.

According to our experiment on face recognition, we conclude that the performance depends more on the problem complexity than the minimum hamming distance of the decomposition matrix, so it is no need to consider seriously the conventional error-correcting concept, and we suggest that the IPC of desired output coding method should be small as one.

References

1. T. Dietterich and G. Bakiri, "Solving Multiclass Learning Problems via Error-Correcting Output Codes", *Journal of Artificial Intelligence Research*, Vol. 2, pp. 263-286, 1995.
2. F. Masulli and G. Valentini, "Effectiveness of Error Correcting Output Codes in Multiclass Learning Problems", *Proc. of the 1st Int'l Workshop on Multiple Classifier Systems*, *Lecture Note in Computer Science*, Vol. 1857, pp. 107-115, 2000.
3. T. Windeatt and R. Ghaderi, "Coding and decoding strategies for multi-class learning problems", *Information Fusion*, Vol. 4, pp. 11-21, 2003.
4. G. Rassch and A. Smola, "Adapting Codes and Embeddings for Polychotomies", *Advances in Neural Information Processing Systems*, Vol. 15, 2003.
5. G. James and T. Hastie, "The Error Coding Method and PICTs", *Computational and Graphical Statistics*, Vol. 7, pp. 337-387, 1998.
6. J. Furnkranz, "Round Robin Rule Learning", *Proc. of the 18th Int'l Conf. on Machine Learning*, pp. 146-153, 2001.
7. V. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.
8. K. Tumar and J. Gosh, "Error Correlation and Error Reduction in Ensemble Classifier", *Tech. Report*, Dept. of ECE, Univ. Texas, July 11, 1996.
9. E. Allwein, R. Schapire and Y. Singer, "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers", *Journal of Machine Learning Research*, Vol. 1, pp. 113-141, 2000.
10. M. Moreira and E. Mayoraz, "Improved Pairwise Coupling Classification with Correcting Classifiers", *Proc. of European Conf. on Machine Learning*, pp. 160-171, 1998.
11. J. Ghosh, "Multiclassifier Systems: Back to the Future", *Proc. of the 3rd Int'l Workshop on Multiple Classifier Systems*, *Lecture Note in Computer Science*, Vol. 2364, pp. 1-15, 2002.
12. T. Hastie and R. Tibshirani, "Classification by Pairwise Coupling", *Advances in Neural Information Processing Systems*, Vol. 10, pp. 507-513, MIT Press, 1998; *The Annals of Statistics*, Vol. 26, No. 1, pp. 451-471, 1998.
13. P. Phillips, H. Moon, S. Rizvi and P. Rauss, "The FERET evaluation methodology for face-recognition algorithms", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22, No. 10, pp. 1090-1104, 2000.
14. M. Almedia, "SMOBR-A SMO program for training SVMs", Dept. of EE, Univ. of Minas Gerais, 2000, Available: <http://www.litc.cpdee.ufmg.br/~barros/svm/smobr>.
15. J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines", *Tech. Report 98-14*, Microsoft Research, Redmond, 1998.

Incremental Classification Rules Based on Association Rules Using Formal Concept Analysis

Anamika Gupta, Naveen Kumar, and Vasudha Bhatnagar

Department of Computer Science,
University of Delhi, India

Abstract. Concept lattice, core structure in Formal Concept Analysis has been used in various fields like software engineering and knowledge discovery. In this paper, we present the integration of Association rules and Classification rules using Concept Lattice. This gives more accurate classifiers for Classification. The algorithm used is incremental in nature. Any increase in the number of classes, attributes or transactions does not require the access to the previous database. The incremental behavior is very useful in finding classification rules for real time data such as image processing. The algorithm requires just one database pass through the entire database. Individual classes can have different support threshold and pruning conditions such as criteria for noise and number of conditions in the classifier.

Keywords: Classification rules, Formal concept analysis, Data Mining, Concept lattice.

1 Introduction

Data Mining can be described as a process of exploration and analysis of large data sets in order to discover meaningful patterns and rules. Data Mining involves scientists from a wide range of disciplines, including mathematicians, computer scientists and statisticians, as well as those working in fields such as machine learning, artificial intelligence, information retrieval and pattern recognition. Classification rule mining and association rule mining are two important data mining techniques. Classification rule mining discovers a small set of rules in the database where consequent of the rule is a class [Q1992][Q1986]. Association rule mining discovers all possible rules in the database that satisfy a user specified minimum support and minimum confidence [AS1994] [SA1996] [PBL1999]. Classification Based on Association rule (CBA) mining aims to find the rules of the form $COND \rightarrow CL$ where COND is the set of conditions and CL is the class label [LHM1998]. Few CBA algorithms have been proposed to address this issue [LHM1998] [HLZC1999]. In this paper we are discussing the method of generating Classification rules Based on Association rules using Concept Lattice of Formal Concept Analysis (CBALattice).

CBALattice provides an efficient method for constructing the concept lattice corresponding to each class label and then it provides an efficient method for building a classifier from the lattice. This method needs only one database pass through the whole procedure. CBALattice can classify data sets irrespective of the number of classes, number of objects (i.e. rows) and number of attributes (i.e. columns). As Association rules deals with whole of data, they give more accurate rules. Since CBALattice is based on association rules so it provides more accurate rules as compared to other traditional methods such as ID3 [Q1986], C4.5 [Q1992]etc.As concepts deal with maximal item sets, concept lattice-based method provides results faster as compared to traditional methods such as ID3, C4.5 etc [HLZC1999] .

CBALattice is incremental in nature. Any increase in number of objects, attributes and classes does not need reading the previous database. Once Classification rules have been generated, concept lattice can be stored. In case of increase in objects, attributes or classes, concepts generated from the incremented database can be combined with concepts stored earlier and new classification rules can be generated.

Traditional association rule mining uses only a single minimum support in rule generation, which is inadequate for unbalanced class distribution. CBALattice method allows us to specify different minimum support for different class label.

Since CBALattice constructs a separate concept lattice for each class label so different pruning conditions can be mentioned for each class label such as criterion of deciding for noise. Also this technique can generate rule, which has many conditions. These rules may be important for accurate classification but it is difficult to find such rules in the CBA methods proposed earlier.

1.1 Related Work

Classification rule mining has been in common use since the emergence of data mining. Several algorithms have been produced such as ID3, C4.5 etc. [LHM1998] proposes an approach of integrating association rules and classification rules. [LHM1998] cannot specify different pruning conditions for each class label except minimum support threshold. Also [LHM1998]is not incremental in nature. It does not consider increase in objects, attributes or classes.

[MAR1996] proposes an algorithm SLIQ, which is scalable. This algorithm can handle large amount of data. But although scalable it is not incremental. Once rules have been generated, any change in the data is not considered. Also SLIQ makes at most two passes over the data for each level of the decision tree. CBALattice makes just one pass of the whole database. SLIQ requires pre-sorting of the data while is not needed in case of our approach.

There are few algorithms present which talk about use of concept lattice in finding Classification rules [HLZC1999] , [LN90], [GMA1995], [XHLL2002], [FFNN2004] , [CR1993] , [MZ2000]. [HLZC1999] gives an algorithm to integrate association rules and classification rules using lattices (CLACF). But it is not incremental. Also it uses same minimum support for all class labels. It cannot

Table 1

	Type of Data	Increase in Classes	Increase in Attributes	Increase in Objects	Lattice Type	Integration of AR and CR	Min sup for different classes	Pruning conditions for different classes
GRAND	Bin	No	No	No	Lattice	No	Same	No
LEGAL	Bin	No	No	Yes	Lattice	No	Same	No
GALOIS	Attr/val	No	No	Yes	Lattice	No	Same	No
RULEARNER	Attr/val	No	No	No	Lattice	No	Same	No
CLNN, CLNB	Symb. Num.	No	No	No	Concept Lattice	No	Same	No
CLACF	Attr/val	No	No	No	Concept Lattice	Yes	Same	No
CBALattice	Bin	Yes	Yes	Yes	Concept Lattice	Yes	Different	Yes

define different pruning conditions for each class label. Concept Lattice produces only maximal item-sets so they are much faster than traditional methods. [HLZC1999] compares apriori and C4.5 with CLACF and found CLACF to be faster than those algorithms.

LEGAL [LM1990] can classify datasets with two classes (positive and negative examples of one concept) only. CBALattice can handle any number of classes. Also CBALattice can handle increase in number of attributes, objects and classes. LEGAL is non-incremental in nature. CLNN and CLNB [XHLL2002] use non-incremental algorithm to build a lattice. GALOIS [GMA1995] generated concepts. It does not generate the rules after that. So objects of test data have to be tested against concepts and not against classification rules. RULEARNER [S1995] uses incremental approach, but it does not make use of Concept Lattice. Our algorithm deals with only concepts, which are less in number.

Table 1 gives a fair idea of comparison of CBALattice with other lattice-based methods. Features of GRAND GALOIS, RULEARNER, CLNN CLNB has been taken from [FFNN2004] .

2 Formal Concept Analysis

Formal Concept Analysis is a field of applied mathematics based on the mathematization of concept and conceptual hierarchy [GW1999]. It thereby activates mathematical thinking for conceptual data analysis and knowledge processing.

A formal context $K = (G, M, I)$ consists of two sets G and M and a relation I between G and M . The elements of G are called the objects and the elements of M are called the attributes of the context. For a set $A \subseteq G$ of objects $A' = \{ m \in M \mid gIm \text{ for all } g \in A \}$ (the set of all attributes common to the objects in A). Correspondingly, for a set B of attributes we define

$B' = \{ g \in G \mid gIm \text{ for all } m \in B \}$ (the set of objects common to the objects in A). A formal concept of the context (G, M, I) is a pair (A, B) with $A \subseteq G, B \subseteq M, A' = B$ and $B' = A$. A is called the extent and B is the intent of the concept (A, B) . $\zeta(G, M, I)$ denotes the set of all concepts of the context (G, M, I) .

If (A_1, B_1) and (A_2, B_2) are concepts of a context, (A_1, B_1) is called a sub concept of (A_2, B_2) , provided that $A_1 \subseteq A_2$ (which is equivalent to $B_2 \subseteq B_1$). In this case, (A_2, B_2) is a super concept of (A_1, B_1) and we write $(A_1, B_1) \leq (A_2, B_2)$. The relation \leq is called the hierarchical order of the concepts. The set of all concepts of (G, M, I) ordered in this way is denoted by $\zeta(G, M, I)$ and is called the concept lattice of the context (G, M, I) .

An ordered set $V: = (V, \leq)$ is a lattice, if for any two elements x and y in V the supremum $x \vee y$ and the infimum $x \wedge y$ always exist. V is called a complete lattice, if the supremum $\bigvee X$ and the infimum $\bigwedge X$ exist for any subset X of V . Every complete lattice V has a largest element, $\bigvee X$, called the unit element of the lattice, denoted by 1_v . Dually, the smallest element 0_v is called the zero elements.

Proposition 1. *Each concept of a context (G, M, I) has the form (X'', X') for some subset $X \subseteq G$ and the form (Y', Y'') for some subset $Y \subseteq M$. Conversely all such pairs are concepts. This implies every extent is the intersection of attribute extents and every intent is the intersection of object intents.*

Using the above proposition this paper gives a method for finding all concepts of the context. Using all concepts now we draw a concept lattice corresponding to each class label.

2.1 Building the Lattice

This algorithm builds Concept Lattice for all class labels. Each class label has a different lattice. The lattice is constructed taking into consideration the minimum support threshold and maximum number of levels. We can prune the rules according to noise. If size of extent of node is less than some threshold then we can consider it as noise. Also while drawing the lattice, we can decide upon the number of levels for the lattice. More levels means more conditions in the rule. This way we can find the rules with desired number of conditions.

```
findConcepts(minSupport,maxlevel,noiseThreshold)
{ for all class labels
  findExtent(classlabel)
  // for all class labels, build a lattice
  // first find all extents of all attributes corresponding to a
  class label
for all attributes
  findExtent(attribute)
  findIntersection(attributeExtent,classlabelExtent,intersect
  Extent)
```

```

if support(intersectExtent) < minSupport
    break;
if (size(intersectExtent)) < noiseThreshold)
    break;
if intersectExtent not present in concept list
    addExtentInConceptList(intersectExtent)
    for all concepts in ConceptList
        findIntersection(conceptExtent,intersectExtent,
            extent)
if extent not present in concept list
    addExtentInConceptList(extent)
endif
addZeroElement()
storeConceptListInFile()
drawLattice(maxLevel)
}

```

For example, consider the following database [HK2001].

	a	b	c	d	e	f	g	h	i	j	CL1	CL2
1.	X				X		X	X				X
2.	X				X		X		X			X
3.		X			X		X	X			X	
4.			X	X			X	X			X	
5.			X	X			X	X			X	
6.	X		X			X			X	X		

Fig. 1

a: age ≤30, b: age=31..40, c: age>40, d: income=low, e: income= medium, f: income=high, g: student=yes, h: student=no, i: credit_rating = fair, j: credit_rating = excellent, Class Label (CL1): buys_computer = yes, CL2: buys_computer = no

Concepts generated for Class Label CL1 from this database are: {36,b}, {45,cehi}, {6,bdgi}, {3,bfhi}, {345,hi}.

Here we have assumed that no support threshold and no limit on level have been specified. Concept lattice:

From the concept lattice we can read the Classification rules, which are based on following proposition:

Proposition 2. *An implication $A \rightarrow B$ holds in (G,M,I) if and only if $B \subseteq A$ ". This means an implication is an Association rule with 100% confidence. Method to read an implication from the concept lattice is as follows: It is sufficient to describe this procedure for implications of the form $A \rightarrow m$, since $A \rightarrow B$ holds if and only if $A \rightarrow m$ holds for each $m \in B$. $A \rightarrow m$ holds if and only if (m',m)*

(A', A'') , i.e. if $\mu m \geq \cap \{ \mu n - n \in A \}$. This means that we have to check in the concept lattice whether the concept denoted by m is located above the infimum of all concepts denoted by an n from A .

2.2 Building the Classifier

Using above Proposition we can now generate Classification rules.

```

findClassifier() { Start from zero element
  go up level by level
  for each branch at all levels
    keep storing the attributes
    last element i.e. unit element is the class label
    (rhs of the rule)
    all attributes connected by 'and' connective is the
    lhs of the rule
  endfor
}

```

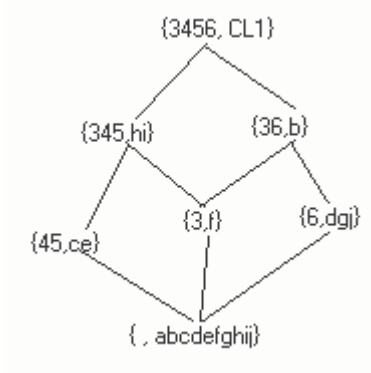


Fig. 2

From the Fig 2, we can find the Classifiers as

1. $c \wedge e \wedge h \wedge i \Rightarrow CL1$
2. $b \wedge f \wedge h \wedge i \Rightarrow CL1$
3. $b \wedge d \wedge g \wedge j \Rightarrow CL1$

2.3 Incremental Classification Rules

Here we have described the algorithm for increase in objects (i.e. the rows). Similarly we can find concepts when increase in attributes or increase in class labels is performed.

```

incrementalLattice(minSupport, maxlevel, noiseThreshold)
{
  if (objectIncremental)
  {
    for all class labels
      findExtent(newClassLabelExtent)
      read oldClassLabelExtent from file
      ClassLabelExtent = oldClassLabelExtent U
      newClassLabelExtent
      readConceptFromFile()
      for all attributes
        findAttributeExtent(newExtent) of incremented context
        readAttributeExtent(oldExtent) from file
        incrementExtent = newExtent U oldExtent
        if support(incrementExtent) < minSupport
          break;
        if (size(intersectExtent)) < noiseThreshold)
          break;
        if incrementExtent not present in concept list
          addExtentInConceptList(incrementExtent)
          for all concepts in ConceptList
            findIntersection(concept, incrementExtent, extent)
            if extent not present in concept list
              addExtentInConceptList(extent)
          endif
      endif
    endfor
  }
}

```

Let's assume that the incremented database is as given in Fig 3.

	a	b	c	d	e	f	g	h	i	j	CL1	CL2
1.	X				X		X	X				X
2.	X				X		X		X			X
3.		X			X		X	X			X	
4.			X	X			X	X			X	
5.			X	X			X	X			X	
6.		X		X			X		X	X		
7.	X			X			X	X			X	
8.			X	X	X		X			X		

Fig. 3

Concepts generated for class label 1 from this incremented database are {7,adig}, {3, bfhi}, {45,cehi}, {3, bfhi}, {36,b}, {458,cei}, {67,dg},{3, bfhi}, {678,g}, {345,hi}, {34578,I}, {6,bdjg}, {8,cegi}, {45,cehi}, {78,gi}

Concept Lattice:

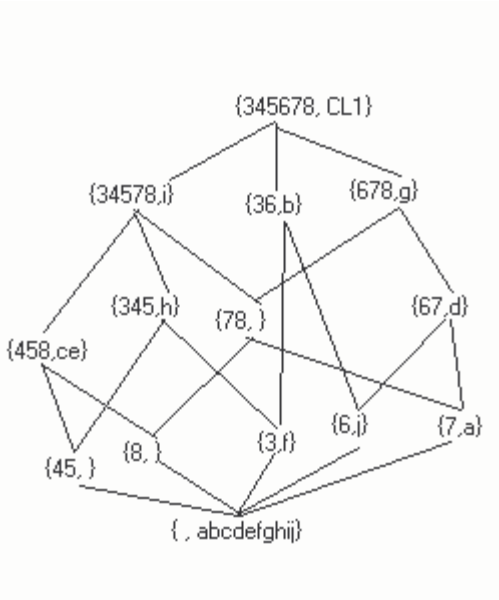


Fig. 4

Few classification rules generated are:

1. $a \wedge d \wedge i \wedge g \Rightarrow CL1$
2. $c \wedge e \wedge i \Rightarrow CL1$
3. $b \wedge f \wedge h \wedge i \Rightarrow CL1$
4. $c \wedge e \wedge h \wedge i \Rightarrow CL1$

3 Experiments

3.1 Computation Effort

The algorithm to find extents of all attributes requires one database pass through the entire database. After finding the extents, we put the extent of class in the list of concepts. Then we find intersection of every attribute extent with the extents in the list of concepts. This requires checking of attribute extent against previously found extents. If intersected extent does not exist in the list then it is put in the list. Effort involved here is checking of intersection of extents. If there are m attributes and n objects then effort required is checking against $m(m+1)/2$ extent list where size of each extent list is less than n . In fact size of each extent list is very less as compared to n . If we increase the number of

attributes to $m+k$ then effort involved in our algorithm is $m(k(k+1)/2)$ whereas if we start the whole process again then effort required is $(m+k)(m+k+1)/2$.

3.2 Accuracy

We have tried our algorithms on dataset from UCI [MM1996] and found that our algorithm gives quite accurate results.

Table 2

Dataset	No of Attrs	No of Binary Attrs	No.- Classes	Size	Error Rate	No. CR with no support	No. CR with support Threshold = 60%
Tic-tac-toe	9	27	2	958	4%	346	20
Zoo	16	36	7	101	4.2%	256	15
Car	6	21	1	1728	6%	3506	436
Hepatitis	19	33	2	155	15%	1692	208

Column 1 denotes the name of the dataset. Column 2 denotes the number of attributes present in the dataset. These attributes may be binary, numeric or continuous. Since CBALattice deals with binary variables only so numeric attributes have been converted to binary attributes. Continuous variables wherever present, have been ignored. Column 3 denotes the number of binary attributes that have been obtained after conversion. In case of Hepatitis dataset, 15 attributes out of 19 attributes have been converted to binary attributes and attributes like Bilirubin, Alk Phosphate, Sgot, Albumin, and Prottime have been ignored (being continuous attributes). Other datasets considered above does not have continuous variables. Column 4 denotes the number of objects present in the dataset. Column 5 denotes the error rate on the datasets. Column 6 denotes the classification rules generated from the concept lattice. Here we have assumed that no support threshold has been mentioned and no other pruning such as number of conditions in the rule, has been performed. If we perform pruning then number of rules generated will be very less. Column 7 denotes the no. of classification rules generated with support threshold = 60%.

4 Conclusion

This paper proposes a framework to integrate association rule and classification rule mining based on concept lattice of formal concept analysis. We propose an algorithm that builds a concept lattice for each class label and then finds the classification rules. The algorithm is incremental in nature. Any increase in number of objects, attributes and classes can be handled very efficiently. Also this algorithm provides a way to define different pruning conditions for different classes.

5 Future Work

CBALattice deals with only binary data. Future version will be able to handle other data. CBALattice can handle large amount of data and since CBALattice is incremental in nature so theoretically it should be scalable also. Scalability can be tested in future.

References

- [AS1994] Agrawal, R. and Srikant, R. 1994, Fast algorithms for mining association rules, VLDB-94, 1994
- [CR1993] Carpineto, C., Romano, G.: Galois: An order-theoretic approach to conceptual clustering. In Proceedings of ICML'93, Amherst (1993) 33-40
- [FFNN2004] Huniyu Fu, Huaiguo Fu, patrik Njiwoua, Engelbert Mephu Nguifo, A Comparative study of FCA- based Supervised Classification of Algorithms, ICFCA 2004.
- [GMA1995] Robert Godin, Missaoui, Hassan Alaoui, Incremental concept formation algorithm based on Galois lattice, Computational Intelligence 11: 246-267 (1995)
- [GW1999] Bernhard Ganter, Rudolf Wille, Formal Concept Analysis, Mathematical Foundations, Springer (1999)
- [HLZC1999] Keyun Hu, Yuchang Lu, Lizhu Zhou, Chunyi Shi, Integrating Classification and Association Rule Mining : A Concept Lattice framework, RSFDGrC 1999, 443-447
- [HK2001] Jiawei Han, Micheline Kamber, Data Mining: Concepts and Techniques
- [LM1990] Liquiere, M., Mephu Nguifo, E. : LEGAL: Learning with Galois lattice.
- [LHM1998] Bing Liu, Wynne Hsu, Yiming Ma, Integrating Classification and Association Rule Mining. In proceedings of KDD-98, 1998.
- [MAR1996] Manish Mehta, Rakesh Agrawal, Jorma Rissanen, SLIQ: A Fast Scalable Classifier for Data Mining, Proc. of the fifth Int'l Conference on Extending Database Technology
- [MM1996] Merz, C.J, and Murthy, P. 1996, UCI repository of machine learning database [<ftp://ftp.ics.uci.edu/pub/machine-learning-databases/>]
- [MZ2000] Mohammed J. Zaki, Generating Non-Redundant Association Rules, In proceedings of the 6th International Conference on Knowledge Discovery and Data Mining (KDD'00).
- [PBL1999] Pasquier N., Bastide Y., Lakhal L. Discovering frequent closed itemsets for association rules. In proceedings of the 7th International Conference on Database Theory (ICDT'99)
- [S1995] Sahami M. : Learning Classification Rules using Lattices, In Proceedings of ECML'95.
- [SA1996] Srikant, R. and Agrawal, R. 1996. Mining quantitative association rules in large relational tables, SIGMOD-96
- [Q1986] Quinlan, J.R. Induction of decision tree. Machine Learning, 1986
- [Q1992] Quinlan, J.R. 1992, C4.5: Program for machine learning, Morgan Kaufmann
- [XHLL2002] Zhipeng XIE, Wynne HSU, Zongtian LIU, Mong Li LEE, Concept Lattice based Composite Classifiers for High Predictability. Journal of Experimental and Theoretical Artificial Intelligence 14 (2002) 143-156

Parameter Inference of Cost-Sensitive Boosting Algorithms

Yanmin Sun¹, A.K.C. Wong¹, and Yang Wang²

¹ Pattern Analysis and Machine Intelligence Lab,
University of Waterloo

{sunym, akcwong}@pami.uwaterloo.ca

² Pattern Discovery Software Ltd
yang.wang@patterndiscovery.com

Abstract. Several cost-sensitive boosting algorithms have been reported as effective methods in dealing with class imbalance problem. Misclassification costs, which reflect the different level of class identification importance, are integrated into the weight update formula of AdaBoost algorithm. Yet, it has been shown that the weight update parameter of AdaBoost is induced so as the training error can be reduced most rapidly. This is the most crucial step of AdaBoost in converting a weak learning algorithm into a strong one. However, most reported cost-sensitive boosting algorithms ignore such a property. In this paper, we come up with three versions of cost-sensitive AdaBoost algorithms where the parameters for sample weight updating are induced. Then, their identification abilities on the small classes are tested on four “real world” medical data sets taken from UCI Machine Learning Database based on F-measure. Our experimental results show that one of our proposed cost-sensitive AdaBoost algorithms is superior in achieving the best identification ability on the small class among all reported cost-sensitive boosting algorithms.

1 Introduction

Reports from both academy and industry indicate that the class imbalance problem has posed a serious drawback of classification performance attainable by most standard learning methods which assume a relatively balanced distribution and equal error cost of the classes [6, 10]. *Class imbalance problem* can be interpreted in two aspects: the *imbalanced class distribution* and the *non-uniform misclassification costs*. Hence, the crucial learning issue is that the class distribution is skewed and the recognition importance on rare events is much higher than that on normal cases. Assuming the balanced class distribution and even recognition importance, traditional learning algorithms do not always produce classifiers which are capable of achieving satisfactory identification performances on rare classes.

AdaBoost (Adaptive Boosting) algorithm, introduced by Freund and Schapire [7, 12], is reported as an effective boosting algorithm to improve classification

accuracy. In view that prevalent classes usually contribute more to the overall classification accuracy, the weighting strategy of AdaBoost may bias towards the prevalent classes. Hence the desired identification ability on small classes is not guaranteed. Cost-sensitive boosting algorithms are therefore developed such that the boosting process may cater to the costly class [5, 13]. However, most reported cost-sensitive boosting algorithm neglect the effects of cost items when choosing the weight update parameter, which is crucial in converting a “weaker” learning algorithm into a strong one.

In this paper, we come up with three versions of cost-sensitive AdaBoost algorithms by inducing the misclassification costs into the weight update formula of AdaBoost in three different ways. For each version, weight update parameter is recalculated taking misclassification costs into consideration. These adaptations retain the good feature of AdaBoost while becoming sensitive to different level of learning importance of different classes. To evaluate their recognition abilities on small classes, four “real world” medical data sets are tested. These data are collections of typical disease diagnostics. Thus, the class imbalance problem prevails in these data sets. *F-measure* evaluation is adopted for performance comparisons.

This paper is organized as follows. Following the introduction in Section 1, section 2 describes the AdaBoost algorithm and addresses the problems of cost-sensitive learning. Section 3 details the methods of integrating misclassification cost into AdaBoost algorithm. Section 4 describes the experimental data, base learner and evaluation measurements. Section 5 compares the recognition abilities of different cost-sensitive boosting algorithms. Section 6 provides the conclusions.

2 AdaBoost and Cost-Sensitive Boosting

2.1 AdaBoost Algorithm

AdaBoost algorithm reported in [7, 12] takes as input a training set $\{(x_1, y_1), \dots, (x_m, y_m)\}$ where each x_i is an n -tuple of attribute values belonging to a certain domain or instance space X , and y_i is a label in a label set Y . In the context of bi-class applications, we can express $Y = \{-1, +1\}$. The Pseudocode for AdaBoost is given as below:

Given: $(x_1, y_1), \dots, (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$

Initialize $D^1(i) = 1/m$.

For $t = 1, \dots, T$:

1. Train base learner h_t using distribution D_t

2. Choose weight updating parameter: α_t

3. Update and normalize sample weights:

$$D^{(t+1)}(i) = \frac{D^{(t)}(i) \exp(-\alpha_t h_t(x_i) y_i)}{Z_t}$$

Where, Z_t is a normalization factor.

Output the final classifier: $H(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x))$

It has been shown in [12] that the training error of the final classifier is bounded as below:

$$\frac{1}{m} |\{i : H(x_i) \neq y_i\}| \leq \frac{1}{m} \sum_i \exp(-y_i f(x_i)) = \prod_t Z_t \quad (1)$$

where,

$$Z_t = \sum_i D^{(t)}(i) \exp(-\alpha_t h_t(x_i) y_i) \quad (2)$$

Minimize Z_t on each round, α_t is induced as

$$\alpha_t = \frac{1}{2} \log \frac{\sum_{i, y_i = h_t(x_i)} D(i)^{(t)}}{\sum_{i, y_i \neq h_t(x_i)} D(i)^{(t)}} \quad (3)$$

To ensure that the selected value of α_t is positive, the following condition should hold

$$\sum_{i, y_i = h_t(x_i)} D(i)^{(t)} > \sum_{i, y_i \neq h_t(x_i)} D(i)^{(t)} \quad (4)$$

2.2 Cost-Sensitive Boosting

Cost-sensitive classification considers varying costs of different misclassification types. Thus the cost-sensitive learning process seeks to minimize the number of high cost errors and the total misclassification cost. Reported works on research in cost-sensitive learning can be categorized into three main groups related to the learning phases of a classifier: 1) Data preprocessing: modifying the distribution of the training set with regards to misclassification costs so that the modified distribution bias towards the costly classes [1, 3]; 2) Classifier Learning: making a specific classifier learning algorithm cost-sensitive [2, 8]; and 3) Classification: using Bayes risk theory to assign each sample to its lowest risk class [4].

Cost-sensitive learning methods in the first group, known as *cost-sensitive learning by example weighting* in [1], is very general since it applies to arbitrary classifier learners and does not change the underlying learning algorithms. In this method, an example-dependent cost is first converted into example weight. Then, a learning algorithm is applied to training examples drawn from this weighted distribution. Several variants of AdaBoost algorithm reported in [5, 13] are with this approach, known as cost-sensitive boosting algorithms.

These cost-sensitive boosting algorithms inherit the learning framework of AdaBoost algorithm. They feed the misclassification costs into the weight update formula of AdaBoost, so that the updated data distribution on the successive boosting round can bias towards the small classes. Except using cost items to update sample weights, CSB1 [13] does not use any α_t factor (or $\alpha_t = 1$), CSB2

[13] uses the same α_t as computed by AdaBoost, and AdaCost [5] introduces a cost adjustment function into weight update rule of AdaBoost. The requirement for this function is: for an instance with a higher cost factor, the function increases its weights “more” if the instance is misclassified, but decreases its weight “less” if otherwise.

The crucial step in AdaBoost algorithm is the selection of the weight update parameter which enables the training error be reduced rapidly. This process is an efficient boosting scheme to convert a weak learning algorithm into a strong one. When introducing the misclassification costs into the weight updating formula, it is necessary to integrate the cost items into the parameter calculation in order to maintain the boosting efficiency. Out of all reported cost-sensitive boosting algorithms, only in AdaCost misclassification costs are taken into consideration when calculating the weight update parameter α . However, the problems with this adaptation are: 1) the selection of the adjustment function is ad hoc; 2) when the cost items (C_P and C_N) are set to 1, AdaCost will not become the original AdaBoost algorithm, thus the steepest descent search of AdaBoost is varied by the cost adjustment function.

3 Cost-Sensitive AdaBoost Algorithms

In order to adapt the weight update strategy of AdaBoost algorithm for cost-sensitive learning, we propose three versions of cost-sensitive AdaBoost algorithms according to the ways we feed the the cost factor into the weight update formula of AdaBoost: inside the exponent, outside the exponent, and both inside and outside the exponent. Let $\{(x_1, y_1, c_1), \dots, (x_m, y_m, c_m)\}$ be a sequence of training samples, where, as denoted previously, each x_i is an n-tuple of attribute values; y_i is a class label in Y where $Y = \{-1, +1\}$, and c_i is the cost factor belonging to the none-negative real domain R^+ . Three modifications of the weight update formula of AdaBoost then become:

$$D^{(t+1)}(i) = \frac{D^{(t)}(i)\exp(-\alpha_t c_i h_t(x_i) y_i)}{Z_t} \quad (5)$$

$$D^{(t+1)}(i) = \frac{c_i D^{(t)}(i)\exp(-\alpha_t h_t(x_i) y_i)}{Z_t} \quad (6)$$

$$D^{(t+1)}(i) = \frac{c_i D^{(t)}(i)\exp(-\alpha_t c_i h_t(x_i) y_i)}{Z_t} \quad (7)$$

Thus, respecting to each modification of weight update formula, a new α value should be calculated to minimize the weighted training error. Taking each modification as a new learning objective, three cost-sensitive AdaBoost algorithms can be developed. We denote them as AdaC1, AdaC2 and AdaC3 respectively. Adopting the inference method used in [12], the calculation of weight updating factor α for each algorithm can be presented in the following subsections.

3.1 AdaC1

Unraveling the weight update rule of Equation 5, we obtain

$$D^{(t+1)}(i) = \frac{\exp(-\sum_t \alpha_t c_i y_i h_t(x_i))}{m \prod_t Z_t} = \frac{\exp(-c_i y_i f(x_i))}{m \prod_t Z_t} \quad (8)$$

where,

$$Z_t = \sum_i D^{(t)}(i) \exp(-\alpha_t c_i y_i h_t(x_i)) \quad (9)$$

Here, the training error bound as stated in Equation 1 still holds. Thus, the learning objective on each round is to find α_t and h_t so as to minimize Z_t (Equation 9). h_t can be trained while minimizing the weighted training error based on current data distribution. Then α_t is selected to minimize Equation 9. According to [12], once $c_i y_i h_t(x_i) \in [-1, +1]$, the following inequality holds

$$\sum_i D(i)^{(t)} \exp(-\alpha c_i y_i h(x_i)) \leq \sum_i D(i)^{(t)} \left(\frac{1 + c_i y_i h_t(x_i)}{2} e^{-\alpha} + \frac{1 - c_i y_i h_t(x_i)}{2} e^{\alpha} \right) \quad (10)$$

By zeroing the first derivative of the right hand side of the inequality (10), α_t can be determined as:

$$\alpha_t = \frac{1}{2} \log \frac{1 + \sum_{i, y_i = h_t(x_i)} c_i D(i)^{(t)} - \sum_{i, y_i \neq h_t(x_i)} c_i D(i)^{(t)}}{1 - \sum_{i, y_i = h_t(x_i)} c_i D(i)^{(t)} + \sum_{i, y_i \neq h_t(x_i)} c_i D(i)^{(t)}} \quad (11)$$

To ensure that the selected value of α_t is positive, the following condition should hold

$$\sum_{i, y_i = h_t(x_i)} c_i D(i)^{(t)} > \sum_{i, y_i \neq h_t(x_i)} c_i D(i)^{(t)} \quad (12)$$

3.2 AdaC2

Unraveling the weight update rule of Equation 6, we obtain

$$D^{(t+1)}(i) = \frac{c_i^t \exp(-\sum_t \alpha_t y_i h_t(x_i))}{m \prod_t Z_t} = \frac{c_i^t \exp(-y_i f(x_i))}{m \prod_t Z_t} \quad (13)$$

where,

$$Z_t = \sum_i c_i D^{(t)}(i) \exp(-\alpha_t y_i h_t(x_i)) \quad (14)$$

Then, the training error of the final classifier is bounded as:

$$\frac{1}{m} |\{i : H(x_i) \neq y_i\}| \leq \frac{1}{m} \sum_i \exp(-y_i f(x_i)) = \prod_t Z_t \sum_i \frac{c_i D^t(i)}{c_i^{t+1}} \quad (15)$$

There exists a constant γ such that $\forall i, \gamma < c_i^{t+1}$. Then,

$$\frac{1}{m} |\{i : H(x_i) \neq y_i\}| \leq \prod_t Z_t \sum_i \frac{c_i D^t(i)}{c_i^{t+1}} \leq \frac{1}{\gamma} \prod_t Z_t \quad (16)$$

Since γ is a constant, the learning objective on each round is to find α_t and h_t so as to minimize Z_t (Equation 14). h_t can be trained while minimizing the weighted training error based on current data distribution. Then α_t is selected to minimize Equation 14 as:

$$\alpha_t = \frac{1}{2} \log \frac{\sum_{i, y_i = h_t(x_i)} c_i D(i)^{(t)}}{\sum_{i, y_i \neq h_t(x_i)} c_i D(i)^{(t)}} \quad (17)$$

To ensure that the selected value of α_t is positive, the following condition should hold

$$\sum_{i, y_i = h_t(x_i)} c_i D(i)^{(t)} > \sum_{i, y_i \neq h_t(x_i)} c_i D(i)^{(t)} \quad (18)$$

3.3 AdaC3

The weight update formula (Equation 7) of AdaC3 is a combination of that of AdaC1 and AdaC2 (with the cost items both inside and outside the exponential function). Then the training error bound of AdaC3 could be expressed as

$$\frac{1}{m} |\{i : H(x_i) \neq y_i\}| \leq \frac{1}{\gamma} \prod_t Z_t \quad (19)$$

where, γ is a constant and $\forall i, \gamma < c_i^{t+1}$, and

$$Z_t = \sum_i c_i D^{(t)}(i) \exp(-\alpha_t c_i y_i h_t(x_i)) \quad (20)$$

Since γ is a constant, the learning objective on each round is to find α_t and h_t so as to minimize Z_t (Equation 20). h_t can be trained while minimizing the weighted training error based on current data distribution. Then α_t is selected to minimize Equation 20.

According to [12], once $c_i y_i h_t(x_i) \in [-1, +1]$, the following inequality holds

$$\sum_i c_i D(i)^{(t)} \exp(-\alpha c_i y_i h(x_i)) \leq \sum_i c_i D(i)^{(t)} \left(\frac{1 + c_i y_i h_t(x_i)}{2} e^{-\alpha} + \frac{1 - c_i y_i h_t(x_i)}{2} e^{\alpha} \right) \quad (21)$$

By zeroing the first derivative of the right hand side of inequality (21), α_t can be determined as:

$$\alpha_t = \frac{1}{2} \log \frac{\sum_i c_i D(i)^{(t)} + \sum_{i, y_i = h_t(x_i)} c_i^2 D(i)^{(t)} - \sum_{i, y_i \neq h_t(x_i)} c_i^2 D(i)^{(t)}}{\sum_i c_i D(i)^{(t)} - \sum_{i, y_i = h_t(x_i)} c_i^2 D(i)^{(t)} + \sum_{i, y_i \neq h_t(x_i)} c_i^2 D(i)^{(t)}} \quad (22)$$

To ensure that the selected value of α_t is positive, the following condition should hold

$$\sum_{i, y_i = h_t(x_i)} c_i^2 D(i)^{(t)} > \sum_{i, y_i \neq h_t(x_i)} c_i^2 D(i)^{(t)} \quad (23)$$

4 Experiment Settings

4.1 Base Learner

To test these cost-sensitive AdaBoost algorithms, we select an associative classification learning system, namely High-Order Pattern and Weigh-of-Evidence Rule Based Classifier (HPWR) as the base learner. The selected base learner HPWR is a complete and independent system. Employing residual analysis and mutual information for decision support, it generates classification patterns and rules in two stages: 1) discovering high-order significant event associations using residual analysis in statistics to test the significance of the occurrence of a pattern candidate against its default expectation[15]; and 2) generating classification rules with weight of evidence attached to each of them to quantify the evidence of significant event associations in support of, or against a certain class membership[14] for a given sample. Hence, HPWR is a mathematically well-developed system with a more comprehensive and rigorous theoretical basis.

4.2 Data Sets

We use four medical diagnosis data sets “Cancer”, “Pima”, “Hypothyroid” and “Sick-euthyroid” taken from UCI Machine Learning Database [11] to test the performances of these three cost-sensitive AdaBoost algorithms. These data sets all have two output labels: one denoting the disease category is treated as the positive class and another representing the normal category is treated as negative class. The percentages of the positive classes are 29.72%, 34.90%, 4.77% and 9.26% respectively.

In these experiments, continuous data in each data set is pre-discretized using the commonly used discretization utility of MLC++ [9] on the default setting and missing values are treated as having the value “?”. Five-fold cross-validation is used on all of the data sets. For consistency, exactly the same data are used to train and test all of these cost-sensitive boosting algorithms.

4.3 Cost Factor

The misclassification costs for samples in the same category are set the same value: C_P denoting the misclassification cost of the positive class and C_N representing that of the negative class. Conceptually, C_P should be greater than C_N . As constrained by the inferences of Inequality 10 and 21, their values should be no greater than 1. Thus, this condition should hold $1 \geq C_P \geq C_N > 0$. In these experiments, relative misclassification costs are set as [1.0 : 1.0, 1.0 : 0.9, 1.0 : 0.8, 1.0 : 0.7, 1.0 : 0.6, 1.0 : 0.5, 1.0 : 0.4, 1.0 : 0.3, 1.0 : 0.2, 1.0 : 0.1] on two classes for all the data sets. Then, with each pair of cost settings, the performance of each learning algorithm is evaluated on 5-fold cross-validation.

4.4 F-Measure for Performance Evaluation

In information retrieval, with respect to a given class, *Recall* is defined as the percentage of retrieved objects that are relevant; and *Precision* is defined as the percentage of relevant objects that are identified for retrieval. Clearly neither of these two measures are adequate by themselves to evaluate the recognition performance on a given class. Thus, the F-measure (F), a measure often-used by the Information Retrieval community for evaluating the performance of the right objects, is devised as a combination of Recall and Precision:

$$F = \frac{2RP}{R + P} \quad (24)$$

It follows that if the F-measure is high when both the recall and precision should be high. This implies that the F-measure is able to measure the “goodness” of a learning algorithm on the current class of interest.

5 Performance Comparisons

Table (1) shows the best F-measure value, as well as the misclassification cost setting of each algorithm on each data set. In general, over the 4 datasets, AdaC3 wins twice, AdaC2 and AdaCost each wins 1 time respectively and AdaC3 also achieves the highest average F-measure value over the four data sets. The performances of CSB1 and CSB2 are obviously not satisfactory.

The basic idea of cost-sensitive boosting algorithm in dealing with the class imbalance problem is to maintain a considerable weighted sample size of the positive class at each iteration of boosting. Then the recognition recall measurement is increased on the positive class. This is the critical step in dealing with the class imbalance problem. However, there is a tradeoff between recall and precision: precision declines as recall increases. When the positive class is over resampled, recall of the positive class is greatly improved. Yet, more samples from the negative class are categorized to the positive class. As a consequence, the recognition precision on the positive class gets worse, and the F-measure

Table 1. F-measure Comparisons of Cost-Sensitive Boosting Algorithms

		HPWR	AdaBoost	AdaC1	AdaC2	AdaC3	AdaCost	CSB1	CSB2
Cancer	Cost			1:0.6	1:0.7	1:0.7	1:0.2	1:0.6	1:0.4
	F_+ (%)	40.21	47.10	50.64	53.98	54.97	50.75	47.88	50.73
Hypo	Cost			1:0.9	1:0.9	1:0.9	1:0.9	1:0.9	1:0.8
	F_+ (%)	55.84	81.99	84.20	84.56	83.02	82.15	82.72	69.42
Pima	Cost			1:0.5	1:0.6	1:0.7	1:0.3	1:0.6	1:0.9
	F_+ (%)	67.98	67.66	72.58	71.03	73.61	69.03	67.30	65.58
Sick	Cost			1:0.8	1:0.8	1:0.9	1:0.9	1:0.8	1:0.8
	F_+ (%)	69.22	79.77	82.51	78.05	81.04	82.67	77.77	62.89
Average F_+ (%)		58.31	69.13	72.48	71.90	73.16	71.24	68.92	57.93

cannot be satisfactory under this situation. Hence, to balance the tradeoff between recall and precision and get the best F-measure value, the boosted weights on the positive class should be in a proper degree which is adequate to obtain a satisfactory recall yet not too much to reduce the precision. Misclassification cost setting is one aspect that influences this issue. Table 1 shows the best ratio setting at which the best F-measure are obtained for each algorithm. Another important affect is related to the weighting scheme of each boosting algorithm. Experimental results reported in Table 1 show that AdaC3 achieves the best F-measure values on two data sets and also the highest average F-measure value over the four data sets.

6 Conclusion

In this paper, we have proposed three new cost-sensitive AdaBoost algorithms to tackle the class imbalance problem in the context of bi-class applications. Based on how cost items are used in the equation, three versions of cost-sensitive boosting algorithms, known as AdaC1, AdaC2 and AdaC3, are developed. To ensure boosting efficiency, α is recalculated taking misclassification costs into consideration for each version. We find that these adaptations retain the good feature of AdaBoost yet adequately sensitive to adjust to cope with different level of learning importance corresponding to different classes. To evaluate their recognition abilities on small classes, four “real world” medical data sets are tested. *F-measure* evaluation is adopted for performance comparisons. In our classification implementation and comparison, we select HPWR as the base learner. When comparing the recognition ability on the small class of each cost-sensitive AdaBoost algorithm, our experimental results show that AdaC3 is superior in achieving the best performance among all reported cost-sensitive boosting algorithms. Further study on weight updating effect of each proposed cost-sensitive boosting algorithm is recommended for theoretically reasoning this observation.

References

1. N. Abe, B. Zadrozny, and J. Langford. An iterative method for multi-class cost-sensitive learning. In Proceedings of the tenth ACN SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 3-11, Seattle, WA, August 2004.
2. J. Bradford, C. Kunz, R. Kohavi, C. Brunk, and C.E. Brodley. Pruning decision trees with misclassification costs. In Proceedings of Tenth European Conference on Machine Learning(ECML-98), pages 131-136, Chemnitz, Germany, April 1998.
3. P. Chan and S. Stolfo. Toward scalable learning with non-uniform class and cost distributions. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pages 164-168, New York, NY, August 1998.
4. C. Elkan. The foundations of cost-sensitive learning. In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, pages 973-978, Seattle, Washington, August 2001.
5. W. Fan, S.J. Stolfo, J. Zhang, and P.K. Chan. Adacost:misclassification costsensitive boosting. In Proc. of Sixth International Conference on Machine Learning(ICML-99), pages 97-105, Bled, Slovenia, 1999.
6. T.E. Fawcett and F. Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291-316, 1997.
7. Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119-139, August 1997.
8. P. Geibel and F. Wyszotzki. Perceptron based learning with example dependent and noisy costs. In T. Fawcett and N. Mishra, editors, Proceedings of the Twentieth International Conference on Machine Learning, pages 218-226. AAAI Press / Mit Press, 2003.
9. R. Kohavi, D. Sommerfield, and J. Dougherty. *Data Mining Using MLC++: A machine learning library in C++*. Tools with Artificial Intelligence. IEEE CS Press, 1996.
10. R. Kubat M., Holte and S. Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30:195-215, 1998.
11. P. M. Murph and D. W. Aha. UCI Repository Of Machine Learning Databases. Dept. Of Information and Computer Science, Univ. Of California: Irvine, 1991.
12. R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297-336, 1999.
13. K.M. Ting. A comparative study of cost-sensitive boosting algorithms. In Proceedings of the 17th International Conference on Machine Learning, pages 983-990, Stanford University, CA, 2000.
14. Y. Wang and A. K. C. Wong. From association to classification: Inference using weight of evidence. *IEEE Trans. On Knowledge and Data Engineering*, 15(3):764-767, 2003.
15. A.K.C. Wong and Y. Wang. High order pattern discovery from discrete-valued data. *IEEE Trans. On Knowledge and Data Engineering*, 9(6):877-893, 1997.

Finite Mixture Models with Negative Components*

Baibo Zhang and Changshui Zhang

State Key Laboratory of Intelligent Technology and Systems,
Department of Automation, Tsinghua University,
Beijing 100084, P.R. China

Abstract. Mixture models, especially mixtures of Gaussian, have been widely used due to their great flexibility and power. Non-Gaussian clusters can be approximated by several Gaussian components, however, it can not always acquire appropriate results. By cancelling the nonnegative constraint to mixture coefficients and introducing a new concept of “negative components”, we extend the traditional mixture models and enhance their performance without increasing the complexity obviously. Moreover, we propose a parameter estimation algorithm based on an iteration mechanism, which can effectively discover patterns of “negative components”. Experiments on some synthetic data testified the reasonableness of the proposed novel model and the effectiveness of the parameter estimation algorithm.

1 Introduction

In the field of statistical learning, finite mixture models (FMM) have been widely used and have continued to receive increasing attention over years due to their great flexibility and power [1]. The capability of representing arbitrary complex probability density functions (pdf’s) enables them to have many applications not only in unsupervised learning fields [2], but also in (Bayesian) supervised learning scenarios and in parameter estimation of class-conditional pdf’s [3]. Especially, Gaussian Mixture Models (GMM) have been widely employed in various applications[1, 2, 3].

GMM can accommodate data of varied structure, since one non-Gaussian component can usually be approximated by several Gaussian ones [4, 5]. However, this approximation can not always acquire appropriate results. To form an intuitive image of this fact, a sample set is generated by a Gaussian model and partly “absorbed” by another one, i.e. there is a “hole” in the data cloud as Fig.1a shows. Fitting this sample set by GMM yields a solution shown in Fig.1b. This solution is achieved by the Competitive Expectation Maximization algorithm (CEM) [6], and the component number is auto-selected by a criterion

* This work is supported by the project (60475001) of the National Natural Science Foundation of China.

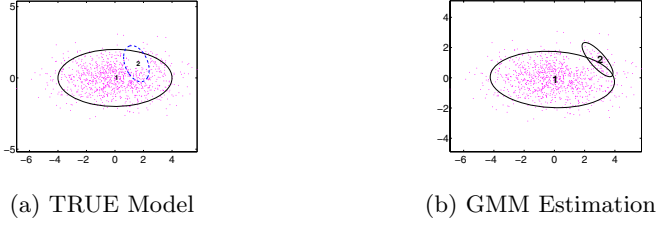


Fig. 1. Samples generated by a component and partly absorbed by another one (average log likelihood in (a) and (b) is 0.353 and 0.351, respectively)

similar to *Minimum Message Length* (MML) [7]. Although this solution is not bad, it is obvious that in the “hole” area , densities are estimated higher than they should be.

In the definition of traditional mixture models, the coefficients of mixture components are nonnegative. In fact, to satisfy the constraint of pdf, it only requires to meet the following two conditions: the sum of the mixture coefficients equals 1, and the probability density at any point is nonnegative. The mixture coefficients are not necessary to be nonnegative.

In this paper, we endeavor to extend mixture models by cancelling the non-negative constraint to mixture coefficients. We introduce a new concept of “Negative Component”, i.e. a component with a negative mixture coefficient.

The rest of this paper is organized as follows. We will describe this proposed model in Sect.2. A parameter estimation algorithm based on an iteration mechanism is given in Sect.3. Experiments are presented in Sect.4, followed by a short discussion and conclusion in Sect.5.

2 Finite Mixture Models with Negative Components

It is said a d -dimensional random variable $x = [x_1, x_2, \dots, x_d]^T$ follows a k -component finite mixture distribution, if its pdf can be written as

$$p(x|\theta) = \sum_{m=1}^k \alpha_m p(x|\theta_m), \quad (1)$$

where α_m is the prior probability of the m th component and satisfies

$$\alpha_m \geq 0, \text{ and } \sum_{m=1}^k \alpha_m = 1. \quad (2)$$

Different descriptions of $p(x|\theta_m)$ can be assigned to different kinds of mixture models. We focus on FMM and demonstrate algorithms by means of GMM.

If the nonnegative constraint of mixture coefficients is cancelled, mixture models will be more powerful to fit data clouds. Equation (2) is modified to

$$\sum_{m=1}^k \alpha_m = 1. \quad (3)$$

To ensure $p(x)$ satisfies the constraint of pdf, we should add a new constraint:

$$p(x) \geq 0, \forall x. \quad (4)$$

For convenience, we call finite mixture models with negative components NegFMM, and call the corresponding GMM version NegGMM.

2.1 An Interpretation to NegFMM

In NegFMM, a component with a positive coefficient is called a ‘‘Positive Component’’ and the negative one a ‘‘Negative Component’’. Let k^+ and k^- denote the number of positive components and negative ones, respectively. $k = k^+ + k^-$ is the total component number. For convenience, positive components and negative ones are separated as follows

$$p(x|\theta) = \sum_{m=1}^{k^+} \alpha_m p(x|\theta_m) + \sum_{m=k^++1}^k \alpha_m p(x|\theta_m) \quad (5)$$

Defining $a = -\sum_{m=k^++1}^k \alpha_m$, $\sum_{m=1}^{k^+} \alpha_m = 1 + a$. Let $\beta_m^+ = \alpha_m/(1 + a)$, $\beta_m^- = -\alpha_{k^++m}/a$. Obviously, $\beta_m^+, \beta_m^- \geq 0$, and $\sum_{m=1}^{k^+} \beta_m^+ = 1$, $\sum_{m=1}^{k^-} \beta_m^- = 1$.

Defining $p^+ = \sum_{m=1}^{k^+} \beta_m^+ p(x|\theta_m)$, $p^- = \sum_{m=1}^{k^-} \beta_m^- p(x|\theta_{k^++m})$, p^+ and p^- are traditional mixture models. So NegFMM can be expressed as

$$p_M = (1 + a)p^+ - ap^-. \quad (6)$$

p^+ is called ‘‘Positive Pattern’’ and p^- is called ‘‘Negative Pattern’’. When $a = 0$, NegFMM will degrade to FMM. In this paper, we only focus on the case of $a > 0$.

Moving p^- to the left side, (6) can be rewritten as

$$p^+ = \frac{1}{1+a} p_M + \frac{a}{1+a} p^-. \quad (7)$$

Then the positive pattern p^+ is expressed as a mixture of the model p_M and the negative pattern p^- . This expression clearly shows that the negative pattern can not exist independently and it is only a part of the positive pattern.

2.2 The Nonnegative Density Constraint for NegGMM

NegFMM introduces the nonnegative density constraint (4). In this section, we will further analyze this constraint in the case of NegGMM.

This constraint is to ensure $(1 + a)p^+ - ap^- = p^+ + a(p^+ - p^-) \geq 0$. When $p^+ - p^- \geq 0$, $p_M \geq 0$ is met. When $p^+ - p^- < 0$, it means

$$a \leq p^+ / (p^- - p^+). \quad (8)$$

We will show that this constraint can be decomposed to two parts, i.e. the constraint to covariance matrices and the constraint to a , corresponding to the nonnegative condition for infinite x and finite x , respectively.

The Covariance Constraint. For Gaussian distribution, the covariance matrix describes the density decaying rate in any direction. For a direction r ($\|r\| = 1$), the variance satisfies $\sigma_r^2 = r^T \Sigma r$, because

$$\sigma_r^2 = \int [r^T(x - \mu)]^2 p(x) dx = r^T \left[\int (x - \mu)(x - \mu)^T p(x) dx \right] r = r^T \Sigma r.$$

For the case of $k^+ = k^- = 1$, if there is a direction r where the variance of p^- is larger than p^+ , the right side of (8) will approach zero when x goes to infinite along the direction r . This will lead to $a = 0$. So the covariance constraint is $\sigma_{1r}^2 \geq \sigma_{2r}^2, \forall r, \|r\| = 1$. Fig.2 illustrate this case: the model in Fig.2a satisfies the covariance constraint while the model in Fig.2b does not.



Fig. 2. Illustration of the covariance constraint

In the general case of NegGMM with arbitrary k^+ and k^- , the constraint will be similar. In any direction, variances of all negative components must be not larger than the maximum variances of all positive components,

$$\max_{1 \leq m \leq k^+} \{\sigma_{mr}^2\} \geq \max_{k^++1 \leq m \leq k} \{\sigma_{mr}^2\}, \quad \forall r, \|r\| = 1. \quad (9)$$

The Constraint to a . If NegGMM satisfies the covariance constraint, there exists a threshold $a_T > 0$. If $a = a_T, \min_x p(x) = 0$. So the constraint to a is

$$a \leq a_T, \quad (10)$$

where $a_T = \min_{x \in \{x | p^- - p^+ > 0\}} \{p^+ / (p^- - p^+)\}$.

3 A Framework of Parameter Estimation

Assuming that samples in the set $X = \{x_1, x_2, \dots, x_n\}$ are independently drawn from the NegGMM model, how to estimate parameters of the model from X is a difficult problem, since no samples originate from the negative pattern.

To estimate an appropriate number of components, many deterministic criteria are proposed [1]. In this paper, we do not consider the problem of choosing the numbers of components. We take the Maximum Likelihood (ML) estimation as our object function,

$$J = \frac{1}{n} \sum_{i=1}^n \log(p(x_i | \theta)). \quad (11)$$

The EM algorithm is a widely used class of iterative algorithms for Maximum Likelihood or Maximum A Posteriori (MAP) estimation in problems with incomplete data, e.g. parameter estimation to mixture models [8, 9]. However, the EM framework is difficult to be directly used in NegGMM, because of the existence of negative coefficients terms.

3.1 Basic Ideas

Parameters of negative pattern p^- can not be directly estimated from the sample set X . According to (7), p^- can be viewed as the result of subtracting p_M from p^+ , where p^+ can be estimated, but p_M is unknown. Intuitively, p_M can be approximated by the sample density function p_s which can be estimated by the Parzen window based methods. Then (7) can be approximated as

$$p^+ = \frac{1}{1+a}p_s + \frac{a}{1+a}p^-. \quad (12)$$

At first p^+ is estimated according to X , then p^- is estimated according to (12). After that a is estimated under the nonnegative density constraint. Then p^+ is reestimated using the information of p^- and a , and so on.

p^+ , p^- and a are optimized separately, i.e. when one part is being optimized, the other parts are fixed. This is similar to the idea of Gibbs Sampling [10].

In order to estimate p^- , we first sampling p^+ to get a sample set. Then, based on (12), we use a modified EM algorithm to estimate p^- with a fixed mixture component p_s .

In order to estimate p^+ , we sampling p^- and weight the sample set according to a . The union of the weighted sample set and X can be viewed as a sample set generated by p^+ . Then p^+ can be estimated by EM.

In order to estimate a , we first estimate the threshold a_T . Then, under the constraint of $a \leq a_T$, we search for the most appropriate a which leads to the highest likelihood value.

The Manifold Parzen Window Algorithm. To estimate p^- , the sample density function p_s needs to be estimated to approximate p_M . To ensure a satisfying result, this estimation should be as accurate as possible. Usually, the sample distribution is inhomogeneous, so the traditional Parzen window method can not promise to obtain a good estimation due to a uniform isotropic covariance.

In this paper, we use the manifold Parzen window algorithm proposed by Vincent [11]. The main idea is that the covariance matrix of sample x_i is calculated by its neighbor points

$$\Sigma_{K_i} = \frac{\sum_{j \neq i} K(x_j; x_i)(x_j - x_i)(x_j - x_i)^T}{\sum_{j \neq i} K(x_j; x_i)}, \quad (13)$$

where the neighbor constraint $K(x; x_i)$ could be soft, e.g. a spherical Gaussian centered on x_i , or hard, e.g. only assigning 1 to the nearest k neighbors and 0 to others. Vincent used the latter in his experiments. Considering the data sparsity in high-dimension space, Vincent added two parameters to enhance

the algorithm, i.e. the manifold dimension d and the noise variance σ^2 . The first d eigenvectors with large eigenvalues to Σ_{K_i} are kept, zeroing the other eigenvalues and then adding σ^2 to all eigenvalues. Based on a criterion of average negative log likelihood, these three parameters are determined by cross validation.

In low-dimension space, Σ_{K_i} is supposed to be nonsingular. So only one parameter, i.e. the neighbor number k , needs to be predetermined. The computational cost for cross validation will be reduced greatly.

Grid Sampling. In order to estimate p^- , we can randomly sampling p^+ . But the randomness will lead to very unstable estimations of p^- because of small number of sampling. To solve this problem, we can increase the amount of sampling which will make the succeeding algorithm very slow, or change random sampling to grid sampling which is adopted in this paper.

For the standard Gaussian model $N(0, I)$, all grid vectors whose lengths are less than d_{scope} will be preserved, and the weight of a grid vector is in proportion to the model density at the point. In our experiments, d_{scope} is determined by experience. The grid space d_{space} can be determined according to precision requirement and computational cost. Let (S_g, W_g) denote the grid set, where S_g are grid vectors and W_g the corresponding grid point weights. Fig.3 shows the Grid sampling for the standard 2D Gaussian model.

For a general Gaussian model $N(\mu, \Sigma)$, where $\Sigma = UAU^T$, the grid sampling set (S, W) is converted from the standard set (S_g, W_g) by $W = W_g$ and $S = UA^{1/2}S_g + \mu \cdot [1, 1, \dots, 1]_{1 \times |S_g|}$.

For traditional Gaussian mixtures, the grid sampling set (S, W) is the union of grid sets of all components, weighting W by component priors once more.

Estimating p^- with one fixed component by EM. EM is widely used to estimate the parameters of mixture models [8, 9]. Our goal is to estimate p^- based on (12). Now we have a sample set (S, W) originating from p^+ (by grid sampling), a component p_s with fixed parameters (estimated using the manifold Parzen window method) and fixed mixture coefficients (determined by a).

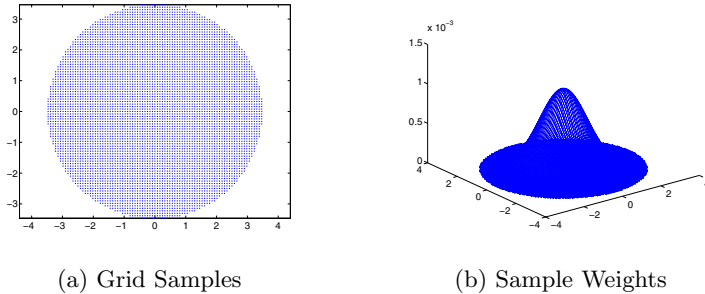


Fig. 3. Grid sampling for standard 2D Gaussian model, $d_{scope}=3.5$, $d_{space}=0.08$

For maximum likelihood estimation, the object function is

$$\sum w_i \ln\left(\frac{a}{1+a} p^-(s_i) + \frac{1}{1+a} p_s(s_i)\right)$$

Similar to the EM algorithm for mixture models[8], the updating formulas can be deduced easily.

E-Step: The posterior to the l th component of p^- can be calculated as

$$p(l|s_i) = \frac{a\beta_l^- p_l^-(s_i)}{ap^-(s_i) + p_s(s_i)}, l = 1, 2, \dots, k^-.$$

This formula is similar to the E-Step in the standard GMM-EM algorithm, except that the denominator contains an additional term $p_s(s_i)$.

M-Step: The updating formulas to the l th component are very similar to the M-Step in the standard GMM-EM algorithm,

$$\beta_l^- = \frac{\sum_i w_{li}}{\sum_{m=1}^{k^-} \sum_i w_{mi}}, \quad \mu_l^- = \frac{\sum_i w_{li} s_i}{\sum_i w_{li}}, \quad \Sigma_l^- = \frac{\sum_i w_{li} (s_i - \mu_l^-)(s_i - \mu_l^-)^T}{\sum_i w_{li}},$$

where w_{li} denotes the weight of s_i to the l th component, and $w_{li} = w_i p(l|s_i)$.

3.2 Scheme of the Parameter Estimation Algorithm

To sum up, the scheme is described as follows:

1. Initialization:

Assign numbers of components k^+ and k^- ;

Estimate sample density function p_s by manifold Parzen window algorithm;

On the sample set X , estimate p^+ using the standard EM algorithm;

Initialize p^- randomly or by k-means based methods on the grid sampling set of p^+ (p_s is used in this step);

Set a to be a small number, e.g. $a = 0.01$, and set iteration counter $t = 0$.

2. One Iteration:

Fixing p^+ and a , estimate p^- by the modified EM algorithm described above;

Fixing p^- and a , estimate p^+ by EM, where the sample set is the union of X (weight is 1) and the grid sample set (S^-, W^-) of p^- (weight is a);

Fixing p^+ and p^- , estimate a under the constraint (10), maximizing (11);

The counter $t = t + 1$.

3. End Condition:

Repeat the iteration 2, until the object function does not change or arrives at the maximal steps. Output θ^* with the maximal J .

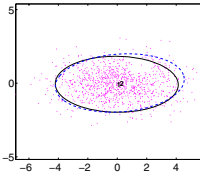
4 Experiments

Example 1. We use 1000 samples from a 2-component 2-dimension NegGMM shown in Fig.1a. The parameters are: $\alpha_1 = 1.05$, $\alpha_2 = -0.05$, $\mu_1 = [0, 0]^T$, $\mu_2 = [1.5, 1]^T$, $\Sigma_1 = \begin{bmatrix} 4 & \\ & 1 \end{bmatrix}$, and $\Sigma_2 = \begin{bmatrix} 0.2 & -0.1 \\ -0.1 & 0.4 \end{bmatrix}$.

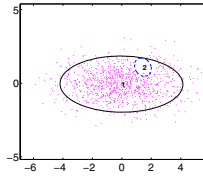
Fig.4 shows the optimization procedure. In this paper, real line and dashed denote positive and negative components respectively, and p^- is initialized by k-means based methods. Fig.4a shows one initial state. Fig.4b~f show some intermediate states of the searching procedure. The best estimation is given in Fig.4f (9th iteration).

Example 2. We use 1000 samples from a 5-component 2-dimension NegGMM shown in Fig.5a, where $k^+ = 2$, $k^- = 3$ and $a = 0.05$. The parameters are:

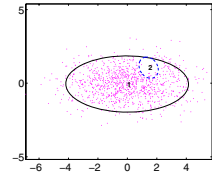
$$\begin{aligned} \alpha_1 &= 0.63, \alpha_2 = 0.42, \alpha_3 = -0.01, \alpha_4 = -0.03, \alpha_5 = -0.01 \\ \mu_1 &= [1.5, 0.2]^T, \mu_2 = [-1.5, 0.3]^T, \\ \mu_3 &= [1.2, 0.4]^T, \mu_4 = [-1.6, 0.4]^T, \mu_5 = [0.2, -0.5]^T \\ \Sigma_1 &= \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.2 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 0.1 & -0.05 \\ -0.05 & 0.1 \end{bmatrix} \\ \Sigma_3 &= \begin{bmatrix} 0.01 & -0.002 \\ -0.02 & 0.02 \end{bmatrix}, \Sigma_4 = \begin{bmatrix} 0.02 & 0.01 \\ 0.01 & 0.02 \end{bmatrix}, \Sigma_5 = \begin{bmatrix} 0.01 & \\ & 0.01 \end{bmatrix} \end{aligned}$$



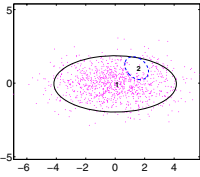
(a) 0, 0.010, 0.337



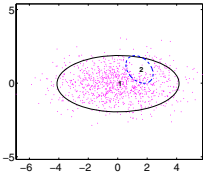
(b) 1, 0.016, 0.349



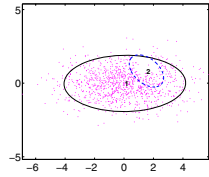
(c) 2, 0.023, 0.352



(d) 3, 0.032, 0.355



(e) 5, 0.047, 0.358



(f) 9, 0.074, 0.359

Fig. 4. Example 1: (a) Initialization (b-e) 1st ~5th iterations (f) the best estimation (values of t , a , J are given below each graph)

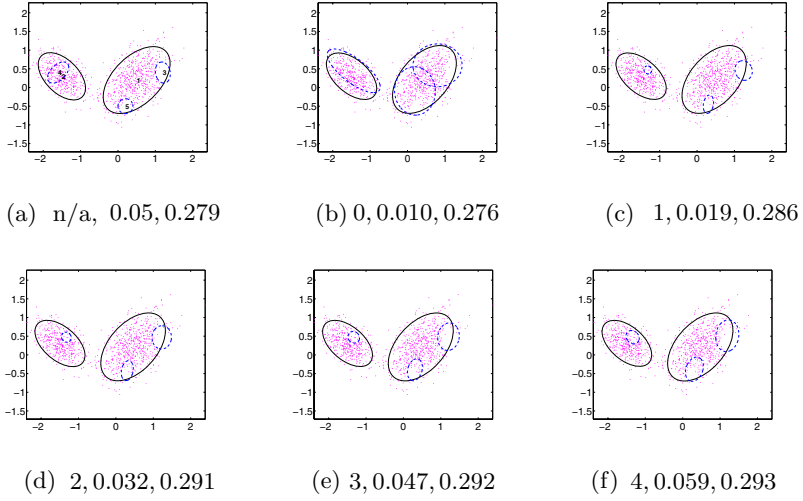


Fig. 5. Example 2: (a) TRUE model (b) Initialization (c-e) $1^{st} \sim 3^{rd}$ iterations (f) the best estimation (values of t, a, J are given below each graph)

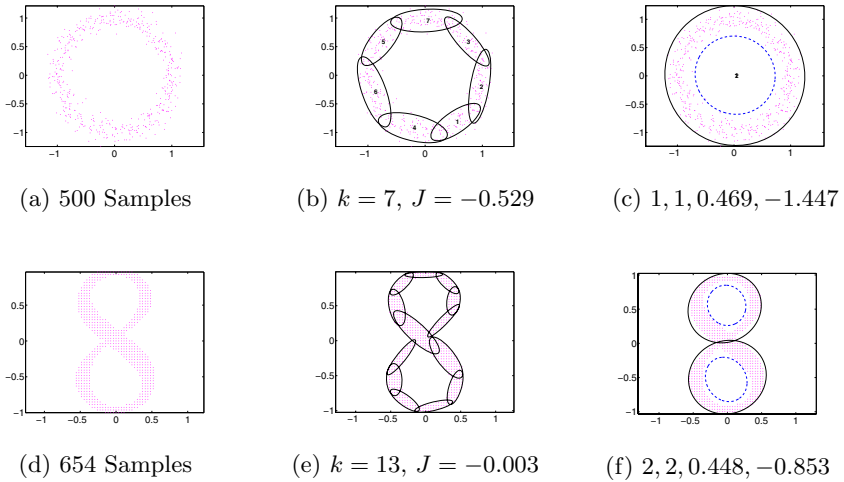


Fig. 6. Some interesting results: (a,d) Sample Sets; (b,e) GMM Estimation by CEM; (c,f) NegGMM Estimation(k^+, k^-, a, J)

Fig.5b~f plot some intermediate states of the searching procedure. The final parameter estimation is given in Fig.5f (4^{th} iteration) where $J = 0.293$. If use the traditional GMM, the best estimation given by CEM is the same as p^+ in the initial state (plotted by real lines in Fig.5b) and the corresponding J equals 0.275.

In our experiments, we do not check the covariance constraint (9). Because the sample set to estimate p^- originates from p^+ , and p^+ contains a fixed component p_s , the estimation of p^- will satisfy the covariance constraint in general. This is also testified by experiments.

For p^- and a , there is observable difference between estimations and true values. It is mainly due to two reasons. The first is the large sampling error between X and the true model (this is also supported by comparing likelihood between the estimation and the true model). The second is that the samples from p^- can not be observed and the estimation algorithm may bring bias.

Some Interesting Examples. Fig.6 illustrates some interesting results. The first column contains sample sets, the second column contains estimations by GMM-CEM where component numbers are auto selected, and the last column contains estimations by NegGMM where component numbers are assigned by us. The first row is a synthetic ring with 500 samples (Fig.6a), the second row is 654 samples drawing from an image of digital “8” (Fig.6d). To traditional GMM, estimations of 7 and 13 components are given respectively (Fig.6b and Fig.6e). These solutions are very good. For NegGMM, estimation results are very interesting (Fig.6c and Fig.6f), though likelihood is lower.

5 Discussion and Conclusion

In this paper, we extend the traditional mixture models by cancelling the nonnegative constraint to mixture coefficients and introduce the concept of “negative pattern”. The power and flexibility of mixture models are enhanced without increasing the complexity obviously.

The proposed parameter estimation framework can effectively discover patterns of lower density relative to positive pattern p^+ due to three tricks. The manifold Parzen window algorithm proposed by Vincent gives a very good estimation of sample density function p_s . The grid sampling helps to gain a very stable estimation of the nonnegative pattern. And the modified EM algorithm gives a final estimation effectively.

Due to the data sparsity, mixture models are difficult to be directly employed in high-dimension space. For the high-dimension case, there are two classes of processing methods. The first is to reduce the data dimension by linear or nonlinear methods, and the second is to constraint the model by priors or hypotheses.

In complex situations, it is very difficult to find an acceptable solution for mixture models by standard EM because of its greed nature. In the future, it is necessary to do more research on split, merge and annihilation mechanism of NegFMM as our previous work[6].

References

1. McLachlan, G., Peel, D.: Finite Mixture Models. New York: John Wiley & Sons (2000)
2. Jain, A.K., Dubes, R.: Algorithm for Clustering Data. Englewood Cliffs, N.J.: Prentice Hall (1988)
3. Hinton, G., Dayan, P., Revow, M.: Modeling the manifolds of images of handwritten digits. *IEEE Trans. on Neural Networks* **8** (1997) 65–74
4. Dasgupta, A., Raftery, A.E.: Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association* **93** (1998) 294–302
5. Fraley, C., Raftery, A.E.: How many clusters? which clustering method? – answers via model-based cluster analysis. *The Computer Journal* **41** (1998) 578–588
6. Zhang, B., Zhang, C., Yi, X.: Competitive EM algorithm for finite mixture models. *Pattern Recognition* **37** (2004) 131–144
7. Figueiredo, M.A., Jain, A.K.: Unsupervised learning of finite mixture models. *IEEE Trans. on PAMI* **24** (2002) 381–396
8. Bilmes, J.A.: A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden markov models. Technical Report ICSI TR-97-021, UC Berkeley (1997)
9. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum-likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistic, Society B* **39** (1977) 1–38
10. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on PAMI* **6** (1984) 721–741
11. Vincent, P., Bengio, Y.: Manifold Parzen windows. In: NIPS. (2002)

MML-Based Approach for Finite Dirichlet Mixture Estimation and Selection

Nizar Bouguila and Djemel Ziou

Département d'Informatique, Faculté des Sciences,
Université de Sherbrooke,
Sherbrooke, Qc, Canada J1K 2R1
{nizar.bouguila, djemel.ziou}@usherbrooke.ca

Abstract. This paper proposes an unsupervised algorithm for learning a finite Dirichlet mixture model. An important part of the unsupervised learning problem is determining the number of clusters which best describe the data. We consider here the application of the Minimum Message length (MML) principle to determine the number of clusters. The Model is compared with results obtained by other selection criteria (AIC, MDL, MMDL, PC and a Bayesian method). The proposed method is validated by synthetic data and summarization of texture image database.

1 Introduction

Statistical models are widely used in various fields such as image processing, pattern recognition, machine learning and remote sensing [1]. In these models, data is characterized in terms of its likely behavior, by means of a probability. The performance of the resulting algorithms depends heavily on the accuracy of the probabilistic models employed. Among the probability models, finite mixtures of densities are widely used [2]. Finite mixtures of distributions are a flexible and powerful modeling which has provided a mathematical based approach to the statistical modeling of a wide variety of random phenomena. This makes them an excellent choice in Bayesian learning. In statistical pattern recognition, finite mixtures permit a formal approach to unsupervised learning. The adoption of this model-based approach to clustering brings important advantages: for instance, the selection of the number of clusters or the assessment of the validity of a given model can be addressed in a formal way. Indeed, an important part of the modeling problem concerns determining the number of consistent components which best describes the data. For this purpose, many approaches have been suggested, such as the Minimum Message Length (MML) [3], Akaike's Information Criterion (AIC) [4], the Minimum Description Length (MDL) [5], the MMDL [6] and the partition coefficient (PC) [7]. Besides, many Bayesian model selection approaches was proposed such as the model of Roberts et al. [8].

In this paper, we consider MML and Dirichlet mixtures. MML has been used especially in the case of Gaussian, Poisson, Von Miss circular mixtures [9] and recently in the case of Gamma [10] mixtures. However, we have proven in

a previous work that the Dirichlet may provide a better fit [11] [12]. 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 [13].

Let us consider a set of data $\mathcal{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N)$ controlled by a mixture of distributions with parameters $\Theta = (\theta_1, \theta_2, \dots, \theta_M)$, where M is the number of clusters, and θ_j is a vector which contains the parameters of the j^{th} distribution. According to information theory, the optimal number of clusters of the mixture is that which allows a minimum amount of information, measured in nats, needed to transmit \mathcal{X} efficiently from a sender to a receiver. The message length is defined as $MessLen = -\log(P(\Theta|\mathcal{X}))$. The minimum message length principle has strong connections with Bayesian inference, and hence uses an explicit prior distribution over parameter values [9]. Baxter [9] gives us the formula for the message length for a mixture of distributions:

$$MessLen \simeq -\log(h(\Theta)) - \log(p(\mathcal{X}|\Theta)) + \frac{1}{2} \log(|F(\Theta)|) + \frac{N_p}{2} (1 - \log(12)) \quad (1)$$

where $h(\Theta)$ is the prior probability, $p(\mathcal{X}|\Theta)$ is the likelihood, and $|F(\theta)|$ is the Fisher information, defined as the determinant of the Hessian matrix of minus the log-likelihood of the mixture. N_p is the number of parameters to be estimated. The estimation of the number of clusters is carried out by finding the minimum with regards to Θ of the message length $MessLen$. In dimension dim , the Dirichlet pdf is defined by:

$$p(\mathbf{X}|\boldsymbol{\alpha}) = \frac{\Gamma(|\boldsymbol{\alpha}|)}{\prod_{i=1}^{dim+1} \Gamma(\alpha_i)} \prod_{i=1}^{dim+1} X_i^{\alpha_i-1} \quad (2)$$

where $\sum_{i=1}^{dim} X_i < 1$, $|\mathbf{X}| = \sum_{i=1}^{dim} X_i$, $0 < X_i < 1 \quad \forall i = 1 \dots dim$, $X_{dim+1} = 1 - |\mathbf{X}|$, $|\boldsymbol{\alpha}| = \sum_{i=1}^{dim+1} \alpha_i$, $\alpha_i > 0 \quad \forall i = 1 \dots dim + 1$. This distribution is the multivariate extension of the 2-parameter Beta distribution. A Dirichlet mixture with M components is defined as :

$$p(\mathbf{X}|\Theta) = \sum_{j=1}^M p(\mathbf{X}|\boldsymbol{\alpha}_j) p(j) \quad (3)$$

where $0 < p(j) \leq 1$ and $\sum_{j=1}^M p(j) = 1$. In this case, the parameters of a mixture for M clusters are denoted by $\Theta = (\boldsymbol{\alpha}, \mathbf{P})$, where $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_M)^T$ and $\mathbf{P} = (p(1), \dots, p(M))^T$ is the mixing parameters vector. In the next two sections, we will calculate the Fisher information $F(\Theta)$ and the prior probability density function $h(\Theta)$. Section 4 is devoted to the experimental results.

2 Fisher Information for a Mixture of Dirichlet

Fisher information is the determinant of the Hessian matrix of the logarithm of minus the likelihood of the mixture. In our case, we have a $((M \times (dim + 2)) \times (M \times (dim + 2)))$ Hessian matrix defined by:

$$H_{l_1 l_2} = \frac{\partial^2}{\partial \theta_{l_1} \partial \theta_{l_2}} (-\log p(\mathcal{X}|\theta)) \quad (4)$$

where $l_1 = 1 \dots M \times (dim + 2)$ and $l_2 = 1 \dots M \times (dim + 2)$. The Hessian matrix of a mixture leads to a complicated analytical form of MML which cannot be easily reproduced. We will approximate this matrix by formulating two assumptions, as follows. First, it should be recalled that α and the vector \mathbf{P} are independent because any prior idea one might have about α would usually not be greatly influenced by one's idea about the value of the mixing parameters vector \mathbf{P} . Furthermore, we assume that the components of α are also independent. The Fisher information is then:

$$F(\theta) \simeq F(\mathbf{P}) \prod_{j=1}^M F(\alpha_j) \quad (5)$$

where $F(\mathbf{P})$ is the Fisher information with regards to the mixing parameters of the mixture and $F(\alpha_j)$ the Fisher information with regards to the vector α_j of a single Dirichlet distribution. In what follows we will compute each of these separately. For $F(\mathbf{P})$, it should be noted that the mixing parameters satisfy the requirement $\sum_{j=1}^M p(j) = 1$. Consequently, it is possible to consider the generalized Bernoulli process with a series of trials, each of which has M possible outcomes labeled first cluster, second cluster, ..., M^{th} cluster. The number of trials of the j^{th} cluster is a multinomial distribution of parameters $p(1), p(2), \dots, p(M)$. In this case, the determinant of the Fisher information matrix is:

$$F(P) = \frac{N}{\prod_{j=1}^M p(j)} \quad (6)$$

where N is the number of data elements. For $F(\alpha_j)$, let us consider the j th cluster $\mathcal{X}_j = (\mathbf{X}_l, \dots, \mathbf{X}_{l+n_j-1})$ of the mixture, where $l \leq N$, with parameter α_j . The choice of the j th cluster allows us to simplify the notation without loss of generality. The Hessian matrix when we consider the vector α_j is given by:

$$H(\alpha_j)_{k_1 k_2} = \frac{\partial^2}{\partial \alpha_{j k_1} \partial \alpha_{j k_2}} (-\log p(\mathcal{X}_j|\alpha_j)) \quad (7)$$

where $k_1 = 1 \dots dim + 1$ and $k_2 = 1 \dots dim + 1$. We can write the negative of the log-likelihood function as follows:

$$-\log p(\mathcal{X}_j|\alpha_j) = -\log\left(\prod_{i=l}^{l+n_j-1} p(\mathbf{X}_i|\alpha_j)\right) = -\sum_{i=l}^{l+n_j-1} \log p(\mathbf{X}_i|\alpha_j) \quad (8)$$

We have:

$$-\frac{\partial \log p(\mathcal{X}_j | \boldsymbol{\alpha}_j)}{\partial \alpha_{jk}} = n_j(-\Psi(|\boldsymbol{\alpha}_j|) + \Psi(\alpha_{jk})) - \sum_{i=l}^{l+n_j-1} \log(X_{ik}) \quad (9)$$

Where Ψ is the digamma function. Then,

$$-\frac{\partial^2 \log p(\mathcal{X}_j | \boldsymbol{\alpha}_j)}{\partial \alpha_{jk_1} \partial \alpha_{jk_2}} = -n_j \Psi'(|\boldsymbol{\alpha}_j|) \quad (10)$$

$$-\frac{\partial^2 \log p(\mathcal{X}_j | \boldsymbol{\alpha}_j)}{\partial^2 \alpha_{jk}} = -n_j(\Psi'(|\boldsymbol{\alpha}_j|) - \Psi'(\alpha_{jk})) \quad (11)$$

Where Ψ' is the trigamma function. We remark that $H(\boldsymbol{\alpha}_j)_{k_1 k_2}$ can be written as:

$$H(\boldsymbol{\alpha}_j)_{k_1 k_2} = D + \gamma \mathbf{a} \mathbf{a}^T \quad (12)$$

where $D = \text{diag}[n_j \Psi'(\alpha_{j1}), \dots, n_j \Psi'(\alpha_{j \dim+1})]$, $\gamma = -n_j \Psi'(|\boldsymbol{\alpha}_j|)$, $\mathbf{a}^T = \mathbf{1}$ and $\gamma \neq (\sum_{k=1}^{\dim+1} \frac{a_k^2}{D_{kk}})^{-1}$, then by the theorem (Theorem 8.4.3) given by Graybill [14], the determinant of the matrix $H(\boldsymbol{\alpha}_j)_{k_1 k_2}$ is given by:

$$F(\boldsymbol{\alpha}_j) = (1 + \gamma \sum_{k=1}^{\dim+1} \frac{a_k^2}{D_{kk}}) \prod_{k=1}^{\dim+1} D_{kk} \quad (13)$$

then

$$F(\boldsymbol{\alpha}_j) = (1 - \Psi'(|\boldsymbol{\alpha}_j|) \sum_{k=1}^{\dim+1} \frac{1}{\Psi'(\alpha_{jk})}) n_j^{\dim+1} \prod_{k=1}^{\dim+1} \Psi'(\alpha_{jk}) \quad (14)$$

Once we have the Fisher information for a single Dirichlet distribution, we can use it to calculate the Fisher information for a mixture of Dirichlet distributions. Eq. 5 is rewritten as:

$$F(\boldsymbol{\Theta}) \simeq \frac{N}{\prod_{j=1}^M p(j)} \prod_{j=1}^M (1 - \Psi'(|\boldsymbol{\alpha}_j|) \sum_{k=1}^{\dim+1} \frac{1}{\Psi'(\alpha_{jk})}) n_j^{\dim+1} \prod_{k=1}^{\dim+1} \Psi'(\alpha_{jk}) \quad (15)$$

3 Prior Distribution $h(\boldsymbol{\Theta})$

The performance of the MML criterion is dependent on the choice of the prior distribution $h(\boldsymbol{\Theta})$. Several criteria have been proposed for the selection of prior $h(\boldsymbol{\Theta})$. Following Bayesian inference theory, the prior density of a parameter is either constant on the whole range of its values or the value range is split into cells and the prior density is assumed to be constant inside each cell. Since α and the vector \mathbf{P} are independent, we have:

$$h(\boldsymbol{\Theta}) = h(\alpha) h(\mathbf{P}) \quad (16)$$

We will now define the two densities $h(\alpha)$ and $h(\mathbf{P})$. The \mathbf{P} vector has M dependent components; i.e. the sum of the mixing parameters is one. Thus, we omit one of these components, say $p(M)$. The new vector has $(M - 1)$ independent components. We treat the $p(j)$, $j = 1 \dots M - 1$ as being the parameters of a multinomial distribution. With the $(M - 1)$ remaining mixing parameters, $(M - 1)!$ possible vectors can be formed. Thus, we set the uniform prior density of \mathbf{P} to [15]:

$$h(\mathbf{P}) = \frac{1}{(M - 1)!} \quad (17)$$

For $h(\alpha)$, since α_j , $j = 1 \dots M$ are assumed to be independent:

$$h(\alpha) = \prod_{j=1}^M h(\alpha_j) \quad (18)$$

We will now calculate $h(\alpha_j)$. In fact, we assume that the components of α_j are independent and in the absence of other knowledge about the α_{jk} , $k = 1, \dots, dim + 1$, we use the principle of ignorance by assuming that $h(\alpha_{jk})$ is locally uniform over the range $[0, e^6 \frac{|\hat{\alpha}_j|}{\hat{\alpha}_{jk}}]$ (in fact, we know experimentally that $\alpha_{jk} < e^6 \frac{|\hat{\alpha}_j|}{\hat{\alpha}_{jk}}$), where $\hat{\alpha}_j$ is the estimated vector. We choose the following uniform prior in accordance with Ockham's razor (a simple priors which give good results):

$$h(\alpha_{jk}) = \frac{e^{-6} \hat{\alpha}_{jk}}{|\hat{\alpha}_j|} \quad (19)$$

By substituting Eq. 19 in Eq. 18, we obtain:

$$h(\alpha_j) = \frac{e^{-6(dim+1)}}{|\hat{\alpha}_j|^{dim+1}} \prod_{k=1}^{dim+1} \hat{\alpha}_{jk} \quad (20)$$

and

$$h(\alpha) = \prod_{j=1}^M h(\alpha_j) = e^{-6M(dim+1)} \prod_{j=1}^M \frac{\prod_{k=1}^{dim+1} \hat{\alpha}_{jk}}{|\hat{\alpha}_j|^{dim+1}} \quad (21)$$

So, substituting Eq. 21 and Eq. 17 in Eq. 16, we obtain:

$$\log(h(\Theta)) = - \sum_{j=1}^{M-1} \log(j) - 6M(dim+1) - (dim+1) \sum_{j=1}^M \log(|\hat{\alpha}_j|) + \sum_{j=1}^M \sum_{k=1}^{dim+1} \log(\hat{\alpha}_{jk}) \quad (22)$$

The expression of MML for a finite mixture of Dirichlet distributions is obtained by substituting equations (22) and (15) in equation (1). The complete algorithm of estimation and selection is then as follows:

Algorithm

For each candidate value of M :

1. Estimate the parameters of the Dirichlet mixture using the algorithm in [11] [12].
2. Calculate the associated criterion $MML(M)$ using Eq. 1.
3. Select the optimal model M^* such that:

$$M^* = \arg \min_M MML(M)$$

4 Experimental Results

We compare the results from the MML approach with those obtained using the same model parameters (from the EM algorithm) using other model-order selection criteria/techniques. The methods we compare are the minimum description length (MDL) [5], The MMDL (Mixture MDL)[6], the Akaike's information criterion (AIC) [4], the Partition coefficient (PC) [7] and a Bayesian criterion, which we call B, proposed by Roberts et al. [8].

4.1 Synthetic Data

In the first application we investigate the properties of our model selection on three two-dimensional toy problems. We choose $dim = 2$ purely for ease of representation. In the first example, data were generated from five Dirichlet densities with different parameters. The parameters were: $\alpha_{11} = 10$, $\alpha_{12} = 16$, $\alpha_{13} = 40$, $\alpha_{21} = 23$, $\alpha_{22} = 50$, $\alpha_{23} = 32$, $\alpha_{31} = 15$, $\alpha_{32} = 19$, $\alpha_{33} = 6$, $\alpha_{41} = 29$, $\alpha_{42} = 8$, $\alpha_{43} = 55$, $\alpha_{51} = 60$, $\alpha_{52} = 40$, $\alpha_{53} = 16$. A total of 100 samples for each of densities were taken. The resultant mixture is presented in Fig. 1.a. From table 1, we can see that only the MML found the exact number of clusters. In the second example, data were generated from six Dirichlet densities with different parameters. The parameters were: $\alpha_{11} = 10$, $\alpha_{12} = 16$, $\alpha_{13} = 40$, $\alpha_{21} = 23$, $\alpha_{22} = 50$, $\alpha_{23} = 32$, $\alpha_{31} = 15$, $\alpha_{32} = 19$, $\alpha_{33} = 6$, $\alpha_{41} = 29$, $\alpha_{42} = 8$, $\alpha_{43} = 55$, $\alpha_{51} = 60$, $\alpha_{52} = 40$, $\alpha_{53} = 16$, $\alpha_{61} = 30$, $\alpha_{62} = 30$, $\alpha_{63} = 30$. A

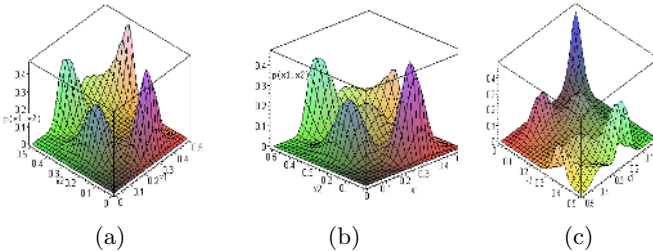


Fig. 1. Mixture densities for the generated data sets

Table 1. Values for the six criteria for the first two-dimensional generated data set

Number of clusters	MML	MDL	AIC	PC	MMDL	B
1	-207.26	-206.16	-401.15	N/A	-206.16	270.41
2	-208.12	-207.02	-401.87	0.63	-207.93	274.45
3	-209.43	-207.89	-401.90	0.76	-209.45	278.84
4	-209.61	-208.00	-403.44	0.75	-210.40	280.13
5	-210.36	-207.54	-401.12	0.70	-210.33	272.02
6	-208.61	-207.01	-400.67	0.67	-211.79	272.98
7	-207.36	-204.43	-399.82	0.65	-209.59	273.17
8	-206.16	-200.12	-398.34	0.66	-207.33	273.91

Table 2. Values for the six criteria for the second two-dimensional generated data set

Number of clusters	MML	MDL	AIC	PC	MMDL	B
1	-287.65	-276.16	-476.52	N/A	-276.16	320.73
2	-288.23	-277.09	-477.09	0.71	-278.31	318.77
3	-288.93	-277.65	-477.54	0.76	-279.20	320.51
4	-289.33	-278.92	-477.78	0.77	-281.32	320.13
5	-289.79	-278.80	-478.33	0.72	-282.29	320.84
6	-290.12	-276.85	-476.97	0.70	-281.65	319.05
7	-287.54	-274.66	-476.80	0.69	-280.11	319.86
8	-287.11	-272.82	-476.66	0.68	-297.80	320.06

Table 3. Values for the six criteria for the third two-dimensional generated data set

Number of clusters	MML	MDL	AIC	PC	MMDL	B
1	-310.18	-300.54	-512.02	N/A	-300.54	378.22
2	-310.87	-300.89	-512.16	0.66	-301.49	380.14
3	-311.22	-301.15	-512.43	0.67	-302.71	379.64
4	-311.93	-301.87	-512.76	0.69	-304.27	379.06
5	-312.37	-302.12	-513.86	0.76	-305.62	378.83
6	-313.37	-303.76	-513.64	0.71	-308.94	380.53
7	-313.55	-301.09	-513.66	0.72	-308.18	379.03
8	-313.49	-300.87	-513.05	0.67	-308.09	379.76

total of 100 samples for each of the fourth first densities and a total of 50 for each of the two last densities were taken. The resultant mixture is presented in Fig. 1.b. From table 2, we can see that only the MML found the exact number of clusters.

In the last example, data were generated from seven densities. The parameters were: $\alpha_{11} = 10$, $\alpha_{12} = 14$, $\alpha_{13} = 40$, $\alpha_{21} = 23$, $\alpha_{22} = 50$, $\alpha_{23} = 32$, $\alpha_{31} = 15$, $\alpha_{32} = 19$, $\alpha_{33} = 6$, $\alpha_{41} = 29$, $\alpha_{42} = 8$, $\alpha_{43} = 55$, $\alpha_{51} = 60$, $\alpha_{52} = 40$, $\alpha_{53} = 16$, $\alpha_{61} = 30$, $\alpha_{62} = 30$, $\alpha_{63} = 30$, $\alpha_{71} = 10$, $\alpha_{72} = 10$, $\alpha_{73} = 40$. A total of 100 samples for each of the three first densities and a total of 50 samples for each of

the four last densities were taken. The resultant mixture is presented in Fig. 1.c. From table 3, we can see that only the MML found the exact number of clusters.

4.2 Real Data

The second application concerns the summarization of image databases. Interactions between users and multimedia databases can involve queries like “Retrieve images that are similar to this image”. A number of techniques have been developed to handle pictorial queries. Summarizing the database is very important because it simplifies the task of retrieval by restricting the search for similar images to a smaller domain of the database. Summarization is also very efficient for browsing. Knowing the categories of images in a given database allows the user to find the images he or she is looking for more quickly. Using mixture decomposition, we can find natural groupings of images and represent each group by the most representative image in the group. In other words, after appropriate features are extracted from the images, it allows us to partition the feature space into regions that are relatively homogeneous with respect to the chosen set of features. By identifying the homogeneous regions in the feature space, the task of summarization is accomplished. For the experiment, we used the *Vistex* grey level texture database obtained from the MIT Media Lab. In our experimental framework, each of the 512×512 images from the *Vistex* database was divided into 64×64 images. Since each 512×512 “mother image” contributes 64 images to our database, ideally all of the 64 images should be classified in the same class. In the experiment, six homogeneous texture groups, “bark”, “fabric”, “food”, “metal”, “water” and “sand” were used to create a new database. A database with 1920 images of size 64×64 pixels was obtained. Four images from each of the bark, fabric and metal texture groups and 6 images from water, food and sand were used. Examples of images from each of the categories are shown in Fig. 2. In order to determine the vector of characteristics for each image, we used the cooccurrence matrix introduced by Haralick et al. [16]. For relevant representation of texture, many cooccurrences should be computed, each one considering a given neighborhood and direction. In our application, we have considered considering the following four neighborhoods : $(1; 0)$, $(1; \frac{\pi}{4})$, $(1; \frac{\pi}{2})$, and $(1; \frac{3\pi}{4})$. For each of these neighborhoods, we calculate the corresponding cooccurrence ma-

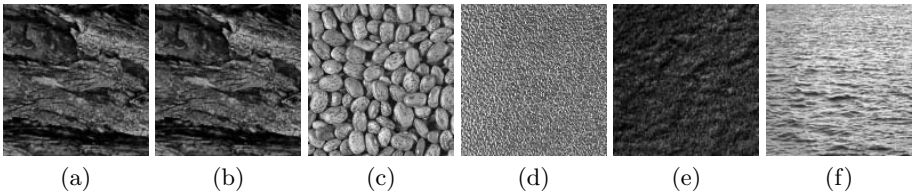


Fig. 2. Sample images from each group. (a) Bark, (b) Fabric, (c) Food, (d) Metal, (e) Sand, (f) Water

Table 4. Number of clusters found by the six criteria

Number of clusters	MML	MDL	AIC	PC	MMDL	B
1	-12945.1	-12951.4	-25643.9	N/A	-12951.4	12543.11
2	-12951.12	-13001.52	-25780.12	0.72	-13002.17	12897.21
3	-12960.34	-13080.37	-25930.23	0.73	-13381.82	12799.54
4	-13000.76	-13206.73	-26000.57	0.82	-13209.81	12730.13
5	-13245.18	-13574.98	-26111.04	0.78	-13578.60	13003.2
6	-13765.04	-13570.09	-26312.64	0.77	-13576.34	13000.11
7	-13456.71	-13493.5	-26401.50	0.74	-13499.53	12761.23
8	-13398.16	-13387.56	-26207.92	0.69	-13393.69	12900.19
9	-13402.64	-13125.41	-26009.95	0.71	-13132.34	12980.32
10	-13100.82	-13001.8	-25999.23	0.80	-13007.81	12580.32

Table 5. Confusion matrix for image classification by a Dirichlet mixture

	Bark	Fabric	Food	Metal	Sand	Water
Bark	250	0	0	0	6	0
Fabric	0	248	8	0	0	0
Food	0	9	375	0	0	0
Metal	0	0	0	250	0	6
Sand	4	0	0	0	380	0
Water	3	0	0	7	2	372

trix, then derive from it the following features: Mean, Energy, Contrast, and Homogeneity. Thus, each image was represented by an $16D$ feature vector. By applying our algorithm to the texture database, only the MML criterion found six categories (see table 4). Then, in what follows we use the selection found by the MML. The classification was performed using the Bayesian decision rule after the class-conditional densities were estimated. The confusion matrix for the texture image classification is given in Table 5. In this confusion matrix, the cell $(class_i, class_j)$ represents the number of images from $class_i$ which are classified as $class_j$. The number of images misclassified was small: 45 in all, which represents an accuracy of 97.65 percent. From table 5, we can see clearly that the errors are due essentially to the presence of macrotexture, i.e the texture at large scale, (between Fabric and food for example) or because of microtexture, i.e the texture at pixel level (between Metal and water for example).

5 Conclusion

We have presented a MML-based criterion to select the number of components in Dirichlet mixtures. The results presented indicate clearly that the MML model selection method which is based upon information theory outperforms the other methods. The validation was based on synthetic data and an interesting applications which involves texture image database summarization.

Aknowledegment

The completion of this research was made possible thanks to the the Natural Sciences and Engineering Research Council of Canada, Heritage Canada and Bell Canada's support through its Bell University Laboratories R&D program.

References

1. A. K. Jain, R. P. W. Duin and J. Mao. Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, January 2000.
2. G.J. McLachlan and D. Peel. *Finite Mixture Models*. New York: Wiley, 2000.
3. C.S Wallace and D.M. Boulton. An Information Measure for Classification. *Computer Journal*, 11(2):195–209, 1968.
4. H. Akaike. A New Look at the Statistical Model Identification. *IEEE Transaction on Automatic Control*, AC-19(6):716–723, 1974.
5. J. Rissanen. Modeling by Shortest Data Description. *Automatica*, 14:465–471, 1987.
6. M. A. T. Frigueiredo, J. M. N. Leitao and A. K. Jain. On Fitting Mixture Models. In E. Hancock and M. Pellilo, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 54–69, 1999.
7. J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, 1981.
8. S.J. Roberts, D. Husmeier, I. Rezek and W. Penny. Bayesian Approaches to Gaussian Mixture Modeling. *IEEE Transactions on PAMI*, 20(11):1133–1142, November 1998.
9. R.A Baxter. *Minimum Message Length Inference: Theory and Applications*. Ph.D. Thesis, Monash University, Clayton, Victoria, Australia, 1996.
10. D. Ziou and N. Bouguila. Unsupervised Learning of a Gamma Finite Mixture Using MML: Application to SAR Image Analysis . In *17th International Conference on Pattern Recognition, ICPR2004*, pages 280–283, 2004.
11. N. Bouguila, D. Ziou and J. Vaillancourt. Unsupervised Learning of a Finite Mixture Model Based on the Dirichlet Distribution and its Application. *IEEE Transactions on Image Processing*, 13(11):1533–1543, November 2004.
12. N. Bouguila, D. Ziou, and J. Vaillancourt. Novel Mixtures Based on the Dirichlet Distribution: Application to Data and Image Classification. In Petra Perner and Azriel Rosenfeld, editors, *Machine Learning and Data Mining in Pattern Recognition*, pages 172–181, 2003.
13. D.L. Dowe and G. Farr. An Introduction to MML Inference. Technical report, Department of Computer Science, Monash University, 1997.
14. F. A. Graybill. *Matrices with applications in Statistics*. Wadsworth, California, 1983.
15. R. A. Baxter and J. J. Olivier. Finding Overlapping Components with MML. *Statistics and Computing*, 10():5–16, 2000.
16. R. M. Haralick, K. Shanmugan and I. Dinstein. Texture features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 8:610–621, 1973.

Principles of Multi-kernel Data Mining*

Vadim Mottl¹, Olga Krasotkina¹, Oleg Seredin¹, and Ilya Muchnik²

¹ Computing Center of the Russian Academy of Sciences,
Vavilov St., 40, 117968 Moscow, Russia
vmottl@yahoo.com

² DIMACS, Rutgers University, P.O. Box 8018, Piscataway, NJ 08855, USA
muchnik@dimacs.rutgers.edu

Abstract. The scientific community has accumulated an immense experience in processing data represented in finite-dimensional linear spaces of numerical features of entities, whereas the kit of mathematical instruments for dissimilarity-based processing of data in metric spaces representing distances between entities, for which sufficiently informative features cannot be found, is much poorer. In this work, the problem of embedding the given set of entities into a linear space with inner product by choosing an appropriate kernel function is considered as the major challenge in the featureless approach to estimating dependences in data sets of arbitrary kind. As a rule, several kernels may be heuristically suggested within the bounds of the same data analysis problem. We treat several kernels on a set of entities as Cartesian product of the respective number of linear spaces, each supplied with a specific kernel function as a specific inner product. The main requirement here is to avoid discrete selection in eliminating redundant kernels with the purpose of achieving acceptable computational complexity of the fusion algorithm.

1 Introduction

The problem of finding empirical dependences $y(\omega) : \Omega \rightarrow Y$ in a set of entities of arbitrary kind $\omega \in \Omega$ is one of the glowing problems of modern data mining. Let a given data set be the set of experimentally measured values of a characteristic $y_j = y(\omega_j) \in Y$ within an accessible subset of entities $\Omega^* = \{\omega_1, \dots, \omega_N\} \subset \Omega$. It is required to continue this function onto the entire set Ω for it would be possible to estimate this characteristic $\hat{y}(\omega)$ for entities $\omega \in \Omega \setminus \Omega^*$ not represented in the original (training) data set [1, 2]. In particular, if $y(\omega)$ takes values from a finite set, for instance, $Y = \{-1, 1\}$, the problem is usually called the pattern recognition problem, and

* This work is supported by the Russian Foundation for Basic Research (Grants 02-01-00107 and 05-01-00679), Grant of the President of the Russian Federation for young scientists No. MK-3173.2004.09 (O. Seredin), INTAS Grant 04-77-7347, and NSF Grant CCR 0325398 (I. Muchnik).

in the case of a real-valued characteristic $Y = \mathbb{R}$ it is referred to as the problem of regression estimation.

It is clear that the problem of function continuation is meaningless until some assumptions are taken about the relations between the values $y(\omega)$ and other characteristics of entities $\omega \in \Omega$ that are more accessible to observation than the goal characteristic. There exist many practical problems of data analysis, including pattern recognition and regression estimation, in which it is relatively easy to evaluate the degree of dissimilarity of any pair of entities. The modern machine learning theory is based on the so-called compactness hypothesis, which consists in the assumption that if two entities are close to each other in the sense of an appropriate metric then so are also, in most cases, the respective values of the goal characteristic. This fact, actually, underlies the featureless (relational, similarity-based) approach to data analysis proposed by R. Duin and his colleagues [3, 4, 5]. In the featureless situation, a natural mathematical model of the general set of entities is a metric space, in which the compactness hypothesis can be expressed directly in terms of the given metric.

At the same time, the mathematically most advanced methods of machine learning essentially exploit the assumption that the universe of entities can be represented as a linear space. As the simplest instrument of introducing linear operations in the set of entities $\omega \in \Omega$, the vector of some observable numerical features $\mathbf{x}(\omega) \in \mathbb{R}^n$ was traditionally considered, and the Euclidean metric produced by it $\rho(\omega', \omega'') = \|\mathbf{x}(\omega') - \mathbf{x}(\omega'')\|$ served as the basis of function continuation in respective machine learning techniques.

It became apparent soon that what immediately determines the result of training is the configuration of the training-set points, represented in \mathbb{R}^n by their pair-wise inner products $(\omega' \cdot \omega'') = \mathbf{x}^T(\omega')\mathbf{x}(\omega'')$, rather than the values of features. This observation resulted in the potential function method of machine learning [2], which later was named the kernel method [1]. The idea of a kernel $K(\omega', \omega'')$ consists in understanding it as inner product of two entities $K(\omega', \omega'') = (\omega' \cdot \omega'')$ in a linear space, maybe, a hypothetical one. If a kernel function $K(\omega', \omega'')$ is defined in an arbitrary set of entities Ω , it produces a Euclidean metric in it

$$\rho(\omega', \omega'') = [K(\omega', \omega') + K(\omega'', \omega'') - 2K(\omega', \omega'')]^{1/2} \quad (1)$$

which expresses a specific compactness hypothesis without the intervening notion of features.

There is usually much freedom in measuring similarity or dissimilarity of entities, and, thus, several heuristic kernels may be suggested within the bounds of the same data analysis problem. However, the choice of features $x_i(\omega) \in \mathbb{R}$, each of which defines, actually, a simplest kernel $K_i(\omega', \omega'') = x_i(\omega')x_i(\omega'')$, is also ever heuristic. The aim of this work is to study the ways of fusing the given set of kernels and to organize, thereby, a concurrence of several compactness hypotheses in finding empirical regularities in the given set of entities. The main requirement here is to avoid discrete selection of kernels with the purpose of achieving acceptable computational complexity of the

fusion algorithm. We use here the main idea of embedding the discrete problem of choosing a subset into a continuous problem of finding optimal nonnegative weights assigned to the elements of the initial set. This idea was originally proposed in [6] as a means of constructing Relevance Vector Machines (RVM).

2 The Linear Space Produced by a Kernel

A kernel $K(\omega', \omega'')$ on a set of entities of arbitrary kind $\omega \in \Omega$ can be defined as a real-valued function $\Omega \times \Omega \rightarrow \mathbb{R}$ possessing two principal properties – symmetry $K(\omega', \omega'') = K(\omega'', \omega')$ and positive semi-definiteness of the matrix $[K(\omega_i, \omega_j); i, j = 1, \dots, m]$ for any finite collection of entities $\{\omega_1, \dots, \omega_m\} \subset \Omega$. The function $\rho(\omega', \omega'')$ (1) produced by a kernel is a metric [7], and, so, the set of entities Ω supplied with a kernel function becomes a metric space.

Any kernel function $K(\omega', \omega'')$ allows for mentally embedding the set Ω into a real linear space with inner product $\Omega \subseteq \tilde{\Omega}$. The null element $\phi \in \Omega$ and linear operations $(\omega' + \omega'' : \tilde{\Omega} \times \tilde{\Omega} \rightarrow \tilde{\Omega})$ and $(c\omega : \mathbb{R} \times \tilde{\Omega} \rightarrow \tilde{\Omega})$ are defined in $\tilde{\Omega}$ in a special way, whereas the role of inner product is played by the kernel function itself $(\omega', \omega'') = K(\omega', \omega'')$.

As the basis for introducing linear operations in the extended set $\tilde{\Omega}$, serves the notion of coaxiality of elements in a metric space [7]. Let $\langle \omega', \omega'' \rangle$ be an ordered pair of elements $\omega', \omega'' \in \tilde{\Omega}$. We shall say that the element $\omega \in \tilde{\Omega}$ is coaxial to the pair $\langle \omega', \omega'' \rangle$ with coefficient $c \in \mathbb{R}$ if $\rho(\omega', \omega) = |c| \rho(\omega', \omega'')$ and $\rho(\omega'', \omega) = |1 - c| \rho(\omega', \omega'')$. This fact will be denoted by the symbol $\omega = \text{coax}(\langle \omega', \omega'' \rangle; c)$. The triangle inequality turns into equality for any three coaxial elements ω', ω'' and ω .

A metric space will be said to be unboundedly convex if for any ordered pair $\langle \alpha', \alpha'' \rangle$ and any $c \in \mathbb{R}$ it contains at least one element coaxial to this pair with coefficient c . It is proved in [7] that the coaxial element is unique if the metric forming an unboundedly convex metric space is produced by a kernel function (1). Such metric spaces are called Euclidean metric spaces. It is assumed that the given set of entities Ω is embedded into a greater set $\Omega \subseteq \tilde{\Omega}$ in which a kernel function is defined and which is, so, a Euclidean metric space.

It is possible to define linear operations in the Euclidean metric space $\tilde{\Omega}$ in the following way (see [7] for details):

- the null element is a hypothetical element $\phi \in \tilde{\Omega}$ for which $K(\phi, \phi) = 0$;
- multiplying by real coefficient $c\omega = \text{coax}(\langle \phi, \omega \rangle; c)$;

- summation $\omega' + \omega'' = 2\text{coax}(\langle \omega', \omega'' \rangle; 1/2)$;
- inner product and norm $(\omega' \cdot \omega'') = K(\omega', \omega'')$, $\|\omega\| = [K(\omega, \omega)]^{1/2}$.

It is just this system of linear operations which is produced in the extended set $\tilde{\Omega}$ by a kernel function defined in the original set of entities $\Omega \subseteq \tilde{\Omega}$.

The dimensionality of the linear space $\tilde{\Omega}$ is the maximum number of elements $\{\omega_1, \dots, \omega_m\} \subset \tilde{\Omega}$ for which the matrix $[K(\omega_i, \omega_j); i, j = 1, \dots, m]$ can be positive definite. We do not study here the question of the dimensionality of this space, which may be finite or infinite, but this issue is extremely important for the generalization performance of the decision rules inferred from a training set.

3 The Class of Linear Decision Rules in the Linear Space Produced by a Kernel Function

The convenience of a kernel function as a means of measuring dissimilarity of any two entities by the respective Euclidean metric (1) consists in that it involves the notion of a linear function $y(\omega) : \Omega \rightarrow \mathbb{R}$ in the set of entities of any kind. This circumstance makes it possible to develop very simple algorithms of estimating dependencies between, generally speaking, arbitrary entities by exploiting, in the featureless situation, practically all known methods which had been worked up for linear spaces.

In this Section, we consider the commonly adopted class of kernel-based decision rules as a class of linear functions in the extended set of entities $\tilde{\Omega}$ supplied with linear operations and inner product produced by a continuation of the given kernel function. The class of linear functions in $\tilde{\Omega}$ is defined by two parameters $\vartheta \in \tilde{\Omega}$ and $b \in \mathbb{R}$

$$y(\omega | \vartheta, b) = K(\vartheta, \omega) + b, \omega \in \Omega. \quad (2)$$

We shall call parameter ϑ the direction element of the linear function.

If the real value of the linear function is immediately treated as the goal characteristic of an entity, the choice of parameters $\vartheta \in \tilde{\Omega}$ and $b \in \mathbb{R}$ determines a regression dependence. If the sign of the linear function is understood as the goal characteristic, the parameters specify a classification of the set of entities into two classes: $y(\omega) = K(\vartheta, \omega) + b > 0 \rightarrow \text{class 1}$, $y(\omega) \leq 0 \rightarrow \text{class 2}$.

Such a way of specifying a linear function may appear nonconstructive because it involves a hypothetical element of a linear space $\vartheta \in \tilde{\Omega}$ as direction element in (2), which is nothing else than product of our imagination. But when solving the problem of inferring the regression dependence or decision rule of pattern recognition from a training set $\{(y_j, \omega_j); j = 1, \dots, N\}$ by the principles of Support Vector Machines [1] or Relevance Vector Machines [6], the only reasonable choice of ϑ will be a linear combina-

tion of really existing objects $\hat{\vartheta} = \sum_{j=1}^N a_j \omega_j$ in accordance with the linear operations induced in the extended set $\tilde{\Omega}$ by the kernel function $K(\omega', \omega'')$ [8]. As inner product in $\tilde{\Omega}$, the kernel function is linear with respect to its arguments, hence, the linear function resulting from training will include the values of the kernel function only for objects existing in reality $\hat{y}(\omega) = \sum_{j=1}^N a_j K(\omega_j, \omega)$.

4 Cartesian Product of Linear Spaces Produced by Several Kernel Functions

It is natural to expect that different experts skilled in the specific knowledge area will propose different kernel functions. The main idea of this work is to shift the burden of the final choice onto the training algorithm by concurrently fusing the given set of heuristically chosen kernels.

Let $K_i(\omega', \omega'')$, $i = 1, \dots, n$, be the kernel functions defined on the same set of entities $\omega \in \Omega$ by different experts. These kernel functions embed the set Ω into different linear spaces $\tilde{\Omega}_i \subset \tilde{\Omega}$, $i = 1, \dots, n$, with different inner products and, respectively, different linear operations. It is convenient to treat the n linear spaces jointly as Cartesian product

$$\tilde{\tilde{\Omega}} = \tilde{\Omega}_1 \times \dots \times \tilde{\Omega}_n = \{ \bar{\omega} = \langle \omega_1, \dots, \omega_n \rangle : \omega_i \in \tilde{\Omega}_i \} \quad (3)$$

formed by ordered n -tuples of elements from $\tilde{\Omega}_1, \dots, \tilde{\Omega}_n$. The kernel function (i.e. inner product) in this linear space can be defined as the sum of the kernel functions (inner products) of the corresponding components in any two n -tuples $\bar{\omega}' = \langle \omega'_1, \dots, \omega'_n \rangle$ and $\bar{\omega}'' = \langle \omega''_1, \dots, \omega''_n \rangle$:

$$K(\bar{\omega}', \bar{\omega}'') = \sum_{i=1}^n K_i(\omega'_i, \omega''_i), \bar{\omega}', \bar{\omega}'' \in \tilde{\tilde{\Omega}}. \quad (4)$$

The dimensionality of the combined linear space $\tilde{\tilde{\Omega}}$ (3) will not exceed the sum of dimensionalities of the particular linear spaces.

A really existing entity $\omega \in \Omega$ will be represented by its n -fold repetition $\bar{\omega} = \langle \omega, \dots, \omega \rangle \in \tilde{\tilde{\Omega}}$. Then any real-valued linear function $\Omega \rightarrow \mathbb{R}$ is specified by the choice of parameters $\bar{\vartheta} \in \tilde{\tilde{\Omega}}$ and $b \in \mathbb{R}$

$$y(\omega) = K(\bar{\vartheta}, \bar{\omega}) + b = \sum_{i=1}^n K_i(\vartheta_i, \omega) + b, \quad (5)$$

where $\bar{\vartheta}$ is a combination of hypothetical elements of particular linear spaces $\bar{\vartheta} = \langle \vartheta_1, \dots, \vartheta_n \rangle$, $\vartheta_i \in \tilde{\Omega}_i$, produced by particular kernel functions $K_i(\omega', \omega'')$ in $\tilde{\Omega}_i$.

Thus, to define a numerical dependence over a set of entities of any kind by combining several kernel functions $K_i(\omega', \omega'')$, we have, first of all, to choose, as parameters, one element in each of linear spaces $\vartheta_i \in \tilde{\Omega}_i$ into which the kernel functions embed the original set $\Omega \subseteq \tilde{\Omega}_i$. It should be marked that the less the norm of the i th parameter in its linear space $\|\vartheta_i\|^2 = K_i(\vartheta_i, \vartheta_i)$, the less the influence of the respective summand on the value of the function (5). If $K(\vartheta_i, \vartheta_i) \rightarrow 0$, i.e. $\vartheta_i \cong \phi_i \in \tilde{\Omega}_i$, the i th kernel will practically not affect the goal function.

This means that the parametric family of numerical functions (5) implies also an instrument of emphasizing “adequate” kernels with respect to the available observations and suppressing “inadequate” ones. Which kernels should be considered as adequate is the key question for providing a good generalization performance of the decision rule when it is applied to entities not represented in the training set.

5 Fusion of Kernel Functions

If the total dimensionality of the combined extended linear space $\tilde{\Omega}$ (3) is greater than the number of entities in the training set $\{(\omega_j, y_j); y_j \in \mathbb{R}, j=1, \dots, N\}$ or $\{(\omega_j, g_j); g_j \in \{-1, 1\}, j=1, \dots, N\}$ there always exist linear functions (5) that exactly reproduce the trainer’s data. Following the widely adopted principle [1], we shall prefer the function with the minimum norm of the direction element under the constraints of the training set:

$$\begin{cases} \|\bar{\vartheta}\|^2 \rightarrow \min, \bar{\vartheta} = \langle \vartheta_1, \dots, \vartheta_n \rangle \in \tilde{\Omega}, \\ \sum_{i=1}^n K_i(\vartheta_i, \omega_j) + b = y_j \\ \text{or } g_j \sum_{i=1}^n K_i(\vartheta_i, \omega_j) + b \geq \text{const.} \end{cases} \quad (6)$$

However, the norm in $\tilde{\Omega}$ can be measured in several ways. The simplest version of norm follows from (4)

$$\|\bar{\vartheta}\|^2 = \sum_{i=1}^n K_i(\vartheta_i, \vartheta_i), \quad (7)$$

but any linear combination of kernel functions with nonnegative coefficients also possesses all the properties of norm $\|\bar{\vartheta}\|^2 = \sum_{i=1}^n (1/r_i) K_i(\vartheta_i, \vartheta_i)$. In this case, the criterion (6) will try to avoid kernels with small r_i . If $r_i = 0$, the respective kernel does not participate in forming the goal function.

The idea of adaptive training consists in jointly inferring the direction elements ϑ_i and the weights r_i from the training set by additionally penalizing large weights [6]:

$$\left\{ \begin{array}{l} \sum_{i=1}^n [(1/r_i)K_i(\vartheta_i, \vartheta_i) + \log r_i] \rightarrow \min(\vartheta_i, r_i), \\ \sum_{i=1}^n K_i(\vartheta_i, \omega_j) + b = y_j \\ \text{or } g_j \sum_{i=1}^n K_i(\vartheta_i, \omega_j) + b \geq \text{const}, j=1, \dots, N. \end{array} \right. \quad (8)$$

This adaptive training criterion displays a pronounced tendency to emphasize the kernel functions which are “adequate” to the trainer’s data and to suppress up to negligibly small values the weights r_i at “redundant” ones.

The reasoning for the adaptive training criterion (8) is the view on the unknown direction elements $\vartheta_i \in \tilde{\Omega}_i$ in each of the linear spaces $\tilde{\Omega}_i$ as hidden independent random variables whose mathematical expectations coincide with the respective null elements $M(\vartheta_i) = \phi_i \in \tilde{\Omega}_i$. The parameter r_i has the sense of the unknown mean-square distance of the random direction element from the null element in the sense of metric (1). Then (8) is equivalent to finding the joint maximum-likelihood estimate of the variables $\vartheta_1, \dots, \vartheta_n$ and their variances r_1, \dots, r_n under the additional assumption that the dimensionality of each of linear spaces $\tilde{\Omega}_i$ is, maybe, very large but finite, and the respective direction element is normally distributed in it.

Since ϑ is element of an abstract linear space but not a vector, for minimizing the Lagrangian of the respective constrained optimization problem (8) we have to use the notion of Frechet differential instead of that of gradient [9]. The Frechet differential of a real-valued function over a linear space is element of this space: $\nabla_{\vartheta} K(\vartheta, \omega) = \omega$, $\nabla_{\vartheta} K(\vartheta, \vartheta) = 2\vartheta$. It can be shown that the following iterative procedure solves both regression estimation and pattern recognition problem:

$$\vartheta_i^{(k)} = r_i^{(k-1)} \sum_{j=1}^N \lambda_j^{(k)} \omega_j \quad \text{or} \quad \vartheta_i^{(k)} = r_i^{(k-1)} \sum_{j=1}^N \lambda_j^{(k)} g_j \omega_j, \quad (9)$$

$$r_i^{(k)} = (r_i^{(k-1)})^2 \sum_{j=1}^N \sum_{l=1}^N K_i(\omega_j, \omega_l) \lambda_j^{(k)} \lambda_l^{(k)}, \quad (10)$$

where the real numbers $\lambda_1^{(k)}, \dots, \lambda_N^{(k)}$ are the Lagrange multipliers (nonnegative in the case of pattern recognition) found as solutions of the respective dual problem. Updating the constant $b^{(k)}$ doesn’t offer any difficulty.

As we see, the abstract variables $\vartheta_i^{(k)} \in \tilde{\Omega}_i$ (9) are linear combinations of the entities of the training set in the sense of linear operations induced by the kernel functions as inner products in the respective linear spaces. Substitution of (9) and (10) into (5) eliminates $\vartheta_i^{(k)}$ and gives the completely constructive estimate of the sought function, respectively, for regression estimation and pattern recognition:

$$\hat{y}^{(k)}(\omega) = \sum_{j=1}^N \lambda_j^{(k)} \sum_{i=1}^n r_i^{(k)} K_i(\omega_j, \omega) + b^{(k)},$$

$$\hat{y}^{(k)}(\omega) = \sum_{j=1}^N \lambda_j^{(k)} g_j \sum_{i=1}^n r_i^{(k)} K_i(\omega_j, \omega) + b^{(k)} \begin{cases} > 0, \\ < 0. \end{cases} \quad (11)$$

As a rule, the process converges in 10-15 steps and displays a pronounced tendency to suppressing the weights at “redundant” kernel functions $r_i \rightarrow 0$ along with emphasizing $r_i \gg 0$ the kernel functions which are “adequate” to the trainer’s data. This fact provides a computationally effective selection of kernel functions without straightforward discrete choice of their subsets.

6 A Particular Case: Feature Selection as Kernel Fusion

There is no insurmountable barrier between the featureless kernel-based way of forming parametric families of numerical functions on a set of entities of any kind and the usual parametric family of linear functions on the set of entities represented by vectors of their numerical features. The latter way is particular case of the former one.

Indeed, a numerical feature $x(\omega): \Omega \rightarrow \mathbb{R}$ is equivalent to the simplest kernel function in the form of product $K(\omega', \omega'') = x(\omega')x(\omega'')$ that embeds the set of entities into a one-dimensional linear space $\Omega \subseteq \tilde{\Omega}$. Respectively, a vector of features $\mathbf{x}(\omega) = [x_1(\omega) \cdots x_n(\omega)]$ gives n kernel functions at once $K_i(\omega', \omega'') = x_i(\omega')x_i(\omega'')$ and n versions of such an embedding $\Omega \subseteq \tilde{\Omega}_i$. The choice of one entity in each of these spaces $\vartheta_i \in \tilde{\Omega}_i$, $i=1, \dots, n$, namely, n real numbers $(x_1(\vartheta_1) \cdots x_n(\vartheta_n)) \in \mathbb{R}^n$, along with a numerical constant $b \in \mathbb{R}$ specifies a linear function on the set of entities: $y(\omega) = \sum_{i=1}^n K_i(\vartheta_i, \omega) + b = \sum_{i=1}^n a_i x_i(\omega) + b$ where $a_i = x_i(\vartheta_i)$.

The less the i th coefficient, i.e. the norm of the i th imaginary entity $\|\vartheta_i\| = x_i(\vartheta_i)$, the less is the contribution of this feature $x_i(\omega)$ to the value of the function.

7 Experimental Results

As the essence of feature selection is shown to be the same as that of kernel fusion, we tested the proposed approach, for obviousness sake, on a set $\{(\mathbf{x}_j, y_j); j=1, \dots, N\}$ of $N=300$ pairs consisting of randomly chosen feature vectors $\mathbf{x}_j \in \mathbb{R}^n$, $n=100$, and scalars obtained by the rule $y_j = a_1 x_{j,1} + a_2 x_{j,2} + \xi_j$

with $a_1 = a_2 = 1$ and ξ_j as normal white noise with zero mean value and some variance σ^2 . So, only $n' = 2$ features of $n = 100$ were rational in the simulated data. In the experiment with regression estimation this set was taken immediately, whereas for the experiment with pattern recognition we took the set $\{(\mathbf{x}_j, g_j); j=1, \dots, N\}$ where $g_j = -1$ if $y_j < 0$ and $g_j = 1$ if $y_j \geq 0$.

In both experiments, we randomly chose $N_{tr} = 20$ pairs for training. So, the size of the training set was ten times greater than the number of rational features, but five times less than the full dimensionality of the feature vector. The remaining $N_{test} = 280$ pairs we used as the test set.

The comparative results of training with equal weights at features (6)-(7) and with adaptive weights (8) are presented in the following two tables:

Regression estimation		
Error rate: ratio of the root-mean-square error in the test set to the actual root variance of the observation noise σ		
Feature set	Training procedure	
	equal weights	adaptive weights
2 rational features	1.01	inapplicable
all 100 features	166.75	2.16

Pattern recognition		
Error rate: misclassification percentage in the test set		
Feature set	Training procedure	
	equal weights	adaptive weights
2 rational features	0.36%	inapplicable
all 100 features	26.8%	0.36%

As was expected, the classical training criterion with equal weights shows a drastic increase in the error rate in both cases when confusing features (i.e. confusing kernel functions) participate in training. At the same time, the error rate with weights adaptation is little sensitive to the presence of purely noisy features. In both experiments, the weights at redundant features turned practically into computer zeros after 10 iterations.

8 Conclusions

A numerical feature, when assigned to entities of a certain kind, embeds the set of these entities into a one-dimensional linear space. The essence of assigning a kernel

function in a set of entities is also embedding it into a hypothetical linear space through the notion of coaxiality of elements of a Euclidean metric space.

The important difference is that the dimensionality of the space induced by a kernel function will be, as a rule, greater than one, if not infinite at all. The main point of the way of fusing several kernels is the idea to consider the Cartesian product of the respective linear spaces, just as the multidimensional feature space formed by a vector of features is the Cartesian product of the respective one-dimensional ones.

Thus, treating the universal set of “all feasible” entities as a linear space practically wipes out the difference between a set of kernels and a set of features and, so, between the featureless and feature-based approach to data analysis. The featureless multi-kernel approach replaces the problem of choosing the features by that of choosing the kernels. According to which of these two problems is easier, the feature-based or the featureless approach should be preferred.

However, fusing too many kernels, just as training with too many features, will inevitably worsen the generalization performance of the decision rule inferred from a small training set unless some regularization measures are taken. The technique of kernel selection proposed here is only one of possible principles of kernel fusion and has its shortcomings. In particular, such a technique should involve elements of testing on a separate set immediately in the course of training, for instance, on the basis of the leave-one-out principle.

References

1. V. Vapnik. *Statistical Learning Theory*. John-Wiley & Sons, Inc. 1998.
2. M.A. Aizerman, E.M. Braverman, L.I. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 1964, Vol. 25, pp. 821-837.
3. Duin, R.P.W, De Ridder, D., Tax, D.M.J. Featureless classification. *Proceedings of the Workshop on Statistical Pattern Recognition*, Prague, June 1997.
4. Duin, R.P.W, De Ridder, D., Tax, D.M.J. Experiments with a featureless approach to pattern recognition. *Pattern Recognition Letters*, vol. 18, no. 11-13, 1997, pp. 1159-1166.
5. Duin, R.P.W, Pekalska, E., De Ridder, D. Relational discriminant analysis. *Pattern Recognition Letters*, Vol. 20, 1999, No. 11-13, pp. 1175-1181.
6. Bishop C.M., Tipping M.E. Variational relevance vector machines. In: C. Boutilier and M. Goldszmidt (Eds.), *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 2000, pp. 46-53.
7. V.V. Mottl. Metric spaces admitting linear operations and inner product. *Doklady Mathematics*, Vol. 67, No. 1, 2003, pp. 140-143.
8. Mottl V., Seredin O., Dvoenko S., Kulikowski C., Muchnik I. Featureless pattern recognition in an imaginary Hilbert space. *Proceedings of the 15th International Conference on Pattern Recognition*. Quebec City, Canada, August 11-15, 2002.
9. Kolmogorov A.N., Fomin S.V. *Introductory Real Analysis*. Prentice-Hall, Englewood Cliffs, 1970.

Comparative Analysis of Genetic Algorithm, Simulated Annealing and Cutting Angle Method for Artificial Neural Networks

Ranadhir Ghosh, Moumita Ghosh, John Yearwood, and Adil Bagirov

School of Information Technology and Mathematical Sciences,
University of Ballarat,
PO Box 663, Ballarat – 3353, Australia
{r.ghosh, m.ghosh, j.yearwood, a.bagirov}@Springer.de

Abstract. Neural network learning is the main essence of ANN. There are many problems associated with the multiple local minima in neural networks. Global optimization methods are capable of finding global optimal solution. In this paper we investigate and present a comparative study for the effects of probabilistic and deterministic global search method for artificial neural network using fully connected feed forward multi-layered perceptron architecture. We investigate two probabilistic global search method namely Genetic algorithm and Simulated annealing method and a deterministic cutting angle method to find weights in neural network. Experiments were carried out on UCI benchmark dataset.

1 Introduction

Artificial neural networks (ANN) are the interconnection of basic units called artificial neurons. Those are capable of performing classification, learning and function approximation. Learning is the main essence of ANN. Learning can be considered as a weight-updating rule of the ANN. Most of the neural learning method strictly depends on the architecture of the ANN. The nonlinearity of ANN results in the existence of many sub-optimal solutions. There are many problems associated with the multiple local minima in neural networks [1][2][3]. Some of the aspects with existing learning methods for MLP can be summarized as the convergence tends to be extremely slow, learning constants must be guessed heuristically, convergence to the global minimum is not guaranteed. [4]. The global search method guarantees the global solution.

There exist solutions that include multiple starts from randomly chosen initial points. Those are simulated annealing, random search, and evolutionary computing [5-14]. These methods are probabilistic in nature and they can find the globally optimal solution with a certain probability. Hence the solution partly depends on the number of iterations of the algorithm. In contrast, there exist deterministic techniques which are capable of finding global optimal solution. Deterministic methods include tabu search, branch-and-bound, generalized cutting plane and systematic search [11,12]. But the computational costs of these methods are extremely high.

In this paper we investigate three different global optimization methods to find the weights of ANN. Two of them are probabilistic global search method namely genetic algorithm, and simulated annealing method respectively. The third one is a recently developed cutting angle method of deterministic global optimization [15-17].

2 Research Methodology

In this section we describe Genetic algorithm, Simulated annealing and Cutting angle method.

2.1 Genetic Algorithm

Genetic algorithm (GA) learning provides an alternative way to learn for the ANN. The task involves controlling the complexity by adjusting the number of weights of the ANN. The use of GA for ANN learning can be viewed as follows:

1. Search for the optimal set of weights
2. Search over topology space
3. Search for optimal learning parameters
4. A combination to search for all the above [18]

The fundamental work in this area was done by Holland, Rechenberg, Schwefel and Fogel during the 1970s [19]. Much of the research has focused on the training of feed forward networks [20] [21]. Miller et al, reported that evolutionary algorithm (EA), which is a slight variation of GA, is a better candidate than other standard neural network techniques, because of the nature of the error surface[22] [23]. Those characteristics pointed out by miller are

1. The architecture surface is infinitely large, hence unbounded for possible number of nodes and connections
2. The surface is non-differentiable since changes in the number of nodes and connections are discrete
3. The surface is complex and noisy since the mapping from the architecture to the performance is indirect, strongly epistasis, and dependent on the evaluation method used.
4. The surface is deceptive since similar architectures may have quite different results
5. The surface is multi-modal since different architectures may have similar performance

The steps in genetic algorithm are described as follows:

Step 1: Initialize all the hidden layer weights using a uniform distribution of a closed interval range of [-1, +1]. A sample genotype for the lower half gene from the population pool for n input, h hidden units, m output, and p number of patterns can be written as $|w_{11}w_{12}...w_{1n}w_{21}w_{22}...w_{2n}...w_{h1}w_{h2}...w_{hn}|$ where, range(w) initially is set between the closed interval [-1 +1]. Also, the sample genotype will vary depending on the connection type as described later.

Step 2: The fitness for the population is calculated based on the phenotype and the target for the ANN.

$$netOutput = f(hid * weight)$$

where **hid** is the output matrix from the hidden layer neurons, **weight** is the weight matrix output neurons and **f** is the sigmoid function

$$RMSError = \sqrt{\frac{\sum_{i=1}^n (netOutput - net)^2}{n * p}}$$

$$popRMSError_i = norm(RMSError_i)$$

norm function normalized the fitness of the individual, so the fitness of each individual population is forced to be within certain range.

Step 3: Generate a random number *x* from a Gaussian distribution of mean 0 and standard deviation 1.

If ($x < crossOverRate$)

Select two parents from the intermediate population

ApplyCrossOver

End If

Generate another random number *y* from the same distribution

If ($y < mutationRate$)

ApplyMutation

End If

The crossover method that is used for this algorithm is known as linear interpolation with two point technique. Let's consider two genes $w_1 w_2 .. w_h$ and $w'_1 w'_2 ... w'_h$.

Two points are selected randomly, lets assume *point1* and *point2*, where $point1 < point2$, and $point1 > 1, point2 < h$.

The two child after the crossover operation will be

$$\frac{2w_1 + w'_1}{3} \frac{2w_2 + w'_2}{3} \dots \frac{2w_{point1} + w'_{point1}}{3} \frac{w_{point1+1} + 2w'_{point1+1}}{3} \dots \frac{w_{point2-1} + 2w'_{point2-1}}{3} \frac{2w_{point2} + w'_{point2}}{3} \dots \frac{2w_h + w'_h}{3}$$

$$\frac{w_1 + 2w'_1}{3} \frac{w_2 + 2w'_2}{3} \dots \frac{w_{point1} + 2w'_{point1}}{3} \frac{2w_{point1+1} + w'_{point1+1}}{3} \dots \frac{2w_{point2-1} + w'_{point2-1}}{3} \frac{w_{point2} + 2w'_{point2}}{3} \dots \frac{w_h + 2w'_h}{3}$$

For mutation, a small random value between 0.1 and 0.2, is added to all the weights. Let us assume a parent string $w_1 w_2 .. w_h$. After mutation the child string becomes $w_1 + \mathcal{E} \mid w_2 + \mathcal{E} \mid .. \mid w_h + \mathcal{E}$, where \mathcal{E} is a small random number [0.1 0.2], generated using a uniform distribution.

Step 4: If the convergence for the GA is not satisfied then goto step 2.2 Manuscript Preparation.

2.2 Simulated Annealing

In this section we will describe the Simulated Annealing method. Let us consider the following global optimization problem:

$$\text{minimize } f(x) \text{ subject to } x \in X \tag{1}$$

where $X \subset R^n$ is a compact set. We describe a version of the simulated annealing (SA) method and its pseudo-code for solving this problem.

Simulated annealing [24-27] is one of the few successful stochastic methods for the practical large-scale problems. Numerical experiments show that SA is successful for many discrete optimization problems. However, for some continuous optimization problems in high-dimensional space SA meets difficulties.

Simulated annealing method differs from the traditional descent methods in that local search algorithm for a neighbourhood solution search allows not only downhill moves, while in an attempt to escape from it allows occasional uphill moves as well. The name “simulated annealing” comes from a physical process called annealing, the process for growing crystals.

Starting with an initial solution x , and an initial “temperature” T_0 , which is a parameter, we obtain a neighbouring solution x' and compare its cost with that of x . If the cost of x' is smaller than that of x , i.e. $f(x') < f(x)$, we accept the new solution x' . The same thing would happen if we were applying the local descent method. On the other hand, if $f(x')$ is greater than $f(x)$ (in which case any local descent algorithm will not accept x'), the SA algorithm may accept x' , but with a probability

$e^{-\frac{\Delta_{x'x}}{T_0}}$ where $\Delta_{x'x}$ is the difference in the costs of x' and x , i.e.

$\Delta_{x'x} = f(x') - f(x)$. This process is carried out for a certain number of times, which we call iterations, for each temperature. Then we reduce the temperature according to a particular schedule, and repeat. An essential element of the SA algorithm

is the probability $e^{-\frac{\Delta_{x'x}}{T}}$ of an uphill move of size $\Delta_{x'x}$ being accepted when the current temperature is T . This is dependent on both $\Delta_{x'x}$ and T . For a fixed temperature T , smaller uphill moves $\Delta_{x'x}$ have a higher probability of being accepted. On the other hand, for a particular uphill move $\Delta_{x'x}$, a higher temperature results in a higher probability for that uphill move to be accepted. In the words of [27], at a high temperature any uphill move might be indiscriminately accepted with a high probability so that the objective function and the tumbles around the space are not very impor-

tant; as T is decreasing the objective function becomes more and more significant; until as T goes to zero the search becomes trapped in the lowest minima that it has reached. Simulated Annealing algorithm for solving a practical problem is typically implemented in two nested loops: the outer loop and the inner loop. The outer loop controls temperatures, while the inner loop iterates a fixed number of times for the given temperature. The inner loop is for the problem of specific decisions. The decisions of the outer loop involve the setting of initial temperature (T_0), the cooling schedule, the temperature length, which is the number of outer loop iterations performed at each temperature, as well as the stopping criterion of the outer loop. The inner loop of SA typically consists of the following parts: feasible solution space, initial feasible solution, neighbourhood move, objective function values, and the decision, which decides whether the decision is found acceptable or probability acceptable according to the so-called Metropolis criterion. Denote *renew* the counts of the solution being accepted in the inner loop, *N_factor* as an input parameter, which can be any positive integer, and *frozen_num* the stopping condition for the outer loop.

The strength of the simulated annealing is that it can deal with highly nonlinear models, chaotic and noisy data and many constraints. It is a robust and general technique. Its main advantages over other local search methods are its flexibility and its ability to approach global optimality. The algorithm is quite versatile since it does not rely on any restrictive properties of the model. The other advantage is that, it allows not only downhill moves while in an attempt to escape from local minima, occasionally it also allow uphill moves. Hence it doesn't get stuck to any narrow or broad local minima and can improve it further.

2.3 Cutting Angle Method

In this section we will describe the Cutting Angle method. The cutting angle method is based on theoretical results in abstract convexity [15]. The method calculates the value of the objective function at certain points. The points are selected in such a way that the algorithm does not return to unpromising regions where function values are high. The new point is chosen where the objective function can potentially take the lowest value. The function is assumed to be Lipschitz, and the value of the potential minim is calculated based on both the distance to the neighboring points and function values at these points. Let us consider the following global optimization problem:

$$\text{minimize } f(x) \text{ subject to } x \in S \quad (2)$$

where the objective function f is an increasing positively homogeneous of degree one and the set S is the unit simplex in R^n .:

$$S = \left\{ x \in R_+^n : \sum_{i=1}^n x_i = 1 \right\} \quad (3)$$

where $R_+^n = \{x \in R^n : x_i \geq 0, i = 1, \dots, n\}$.

A function f defined R_+^n is called increasing if $x \geq y$ implies $f(x) \geq f(y)$. The function f is positively homogeneous of degree one if $f(\lambda, x) = \lambda f(x)$ for all $x \in R_+^n$ and $\lambda > 0$.

For a given vector $l \in R_+^n, l \neq 0$, we consider $I(l) = \{i = 1, \dots, n : l_i > 0\}$. We use the following notation for $c \in R$ and $l \in R_+^n$:

$$(c/l)_i = \begin{cases} c/l_i & \text{if } i \in I(l) \\ 0 & \text{if } i \notin I(l) \end{cases} \quad (4)$$

An IPH function is nonnegative on R_+^n . We assume that $f(x) > 0$ for all $x \in S$. It follows from the positiveness of f that $I(l) = I(x)$ for all $x \in S$ and $l=f(x)/x$. Let e^k be the k th orthant vector.

The cutting angle method is as follows:

Step0: Initialization: Consider the points $x^k \in S, k=1, \dots, m$, where $m \geq n, e^k = x^k$ for $k = 1, \dots, n$ and $x^k \geq 0$ for $k = n + 1, \dots, m$. Let $l^k = f(x^k)/x^k, k = 1, \dots, m$. Define the function h_m :

$$h_m(x) = \max_{k \leq m} \min_{i \in I(l^k)} l_i^k x_i = \max \left\{ \max_{k \leq n} l_k^k x_k, \max_{n+1 \leq k \leq m} \min_{i \in I(l^k)} l_i^k x \right\} \quad (5)$$

And set $j=m$.

Step1: Find a solution x^* for the problem

$$\text{minimize } h_j(x) \text{ subject to } x \in S. \quad (6)$$

Step2: Set $j = j+1$ and $x^j = x^*$.

Step3: Compute $l^j = f(x^j)/x^j$, define the function

$$h_j(x) = \max \left\{ h_{j-1}(x), \min_{i \in I(l^j)} l_i^k x_i \right\} \equiv \max_{k \leq j} \min_{i \in I(l^k)} l_i^k x_i \quad (7)$$

And go to Step 1.

3 Experimental Result

Experiments were conducted using the following real-world benchmark data sets from UCI Machine Learning repository: Austral, Breast cancer (Wisconsin) and Heart Disease (Cleveland) and Diabetes data. The details of these datasets can be obtained from the UCI website. The datasets are described in Table 1.

Table 1. Dataset details

Dataset	Instances	Class	Attribute
Austral	690	2	14
Wisconsin Breast Cancer Databases	699	2	9
Heart Disease Cleveland	297	2	13
Diabetes	768	2	8

The results are compared in terms of test classification accuracy and computation time. The following tables (Table 2 & 3) show the classification accuracy and the time complexity of the ANN in percentage for all methods and data sets.

Table 2. Classification Accuracy results for all data sets

Dataset	Classification Accuracy [%]		
	GA	SA	CA
Austral	88.5	89	92.2
Breast Cancer	96.5	98.8	100
Cleveland	89.7	87.5	89.7
Diabetes	82.3	79.8	81.5

Table 3. Time Complexity results for all data

Dataset	CPU Time [s]		
	GA	SA	CA
Austral	89	75.4	85.6
Breast Cancer	75	69.8	70.3
Cleveland	40	35.5	45
Diabetes	51	46.5	49.8

4 Analysis

The following figures (Figure 1, Figure 2) show a comparison of classification accuracy and the time complexity for the three methods. From Figure 1 it is clear that CA performed more efficiently compare to SA for all datasets. But GA performed slightly better in case of diabetes dataset.

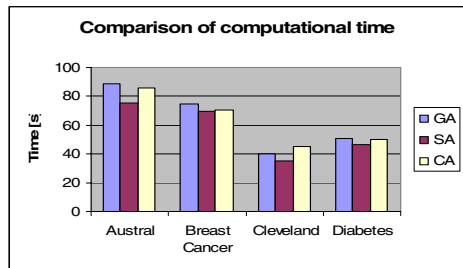
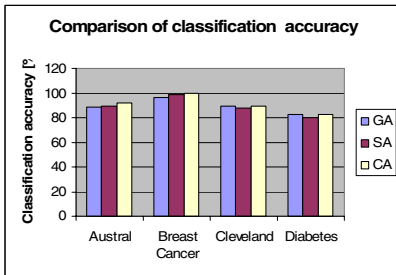


Fig. 1. Comparison of classification accuracy **Fig. 2.** Comparison of computational time

Figure 3 shows the convergence of GA, SA, and CA in astral dataset. Figure 3 shows that SA has converged much quicker than GA and CA. This is because the

stopping criterion of SA was number restricted to number of iteration, because each iteration in SA takes long time to converge. GA has taken much longer time to converge compare to CA.

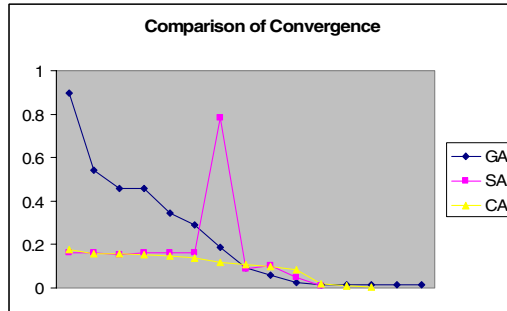


Fig. 3. Comparison of Convergence

5 Conclusion

This paper presents a comparative analysis of probabilistic and deterministic global search method to find neural network weights. The results show that both Cutting angle method, and Genetic algorithm performed much better than Simulated annealing method for all the dataset. While we compare Genetic algorithm with Cutting angle method, we see that that Cutting angle method performed slightly better that Genetic algorithm in most of the cases. For diabetes and Heart Disease dataset Genetic algorithm performed slightly better than Cutting angle method.

References

- [1] Whittle, P.: Prediction and regularization by linear least square methods", Van Nostrand, Princeton, N.J. (1963).
- [2] Goggin, S. D., Gustafson, K.E., and Johnson, K. M.: An asymptotic singular value decomposition analysis of nonlinear multilayer neural networks. International Joint Conference on Neural Networks, (1991), I-785-I-789,.
- [3] Burton, S. A.: A matrix method for optimizing a neural network, Neural comput. Vol 3, no 3.
- [4] Lawrence, S., Giles, C. L., Tsou, A. C.: What size neural network gives optimal generalization? Convergence properties of backpropagation. UMIACS-TR-96-22.
- [5] Duch, W. and Korczak, J.: Optimization and global minimization methods suitable for neural networks. Neural computing surveys (1999).
- [6] Phansalkar V. V. and Thathachar, M. A. L.: Local and Global Optimization Algorithms for Generalized Learning Automata. Neural Computation, Vol. 7. (1995), 950-973.
- [7] Sexton, R., Dorsey R., and Johnson, J.: Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing. European Journal of Operational Research, Vol. 114. (1999), 589-601.

- [8] Sexton, R., Dorsey R., and Johnson, J.: Toward global optimization of neural networks: A comparison of the genetic algorithm and backpropagation. *Decision Support Systems*, Vol. 22. (1998), 171-185.
- [9] Sexton, R., Dorsey R., and Johnson, J.: Beyond Backpropagation: Using Simulated Annealing for Training Neural Networks. *Journal of End User Computing*, Vol. 11. (1999), 3.
- [10] Y. Shang and B. W. Wah, Global optimization for neural network training, *Computer*, 29 (1996), pp. p45(10).
- [11] PintÈr, J.: *Global optimization in action : continuous and Lipschitz optimization--algorithms, implementations, and applications*. Kluwer Academic Publishers, Dordrecht; Boston, (1996).
- [12] Trn, A., and Zhilinskias, A.: *Global optimization*, Springer-Verlag, Berlin ; New York, (1989).
- [13] Zhang, X. M., and Chen, Y. Q.: Ray-guided global optimization method for training neural networks, *Neurocomputing*, Vol. 30. (2000), 333-337.
- [14] Zhang, X.-S.: *Neural networks in optimization*. Kluwer Academic Publishers, Boston, Mass., (2000).
- [15] Rubinov, A. M.: *Abstract convexity and global optimization*. Kluwer Academic Publishers, Dordrecht ; Boston, (2000).
- [16] Andramonov, M., Rubinov, A., and Glover, B.: Cutting angle methods in global optimization. *Applied Mathematics Letters*, Vol. 12 (1999), 95-100.
- [17] Bagirov, A., and Rubinov, A.: Global minimization of increasing positively homogeneous function over the unit simplex, *Annals of Operations Research*, Vol. 98 (2000), 171-187.
- [18] Petridis, V., Kazarlis, S., Papaikonomu, A., and Filelis, A.: A hybrid genetic algorithm for training neural network. *Artificial Neural Networks*, Vol. 2. (1992), 953-956.
- [19] Rechenberg, I.: *Cybernetic solution path of an experimental problem*. Royal Aircraft Establishment, Library translation no. 1122, Farnborough, Hants, U.K, Aug, (1965).
- [20] Whitley, D., Starkweather, T., and BoEArt, C. Genetic algorithms and neural networks - optimizing connections and connectivity. *Parallel Computing*, Vol. 14, (1990). 347-361.
- [21] Montana, D. , and Davis, L.: Training feedforward neural networks using genetic algorithms. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence IJCAI-89*, Vol. 1, (1989).
- [22] Frean, M.: The upstart algorithm: a method for constructing and training feedforward neural networks. *Neural computation*, Vol. 2, (1990).
- [23] Roy, A., Kim, L.S., and Mukhopadhyay, S.: A polynomial time algorithm for the construction and training of a class of multiplayer perceptrons. *Neural networks*, Vol. 6, (1993).
- [24] Hedar, A.R. and Fukushima, M.: Hybrid Simulated Annealing and Direct Search method for nonlinear unconstrained global optimization. *Optimization Methods and Software*, Vol. 17, No. 5, (2002). 891-912.
- [25] Brooks, D. G. and Verdini, W.A.: Computational experience with generalized simulated annealing over continuous variables. *American Journal of Mathematical and Management Sciences*, Vol. 8. (1988). 425-449.
- [26] Cardoso, M. F., Salcedo, R. L., and de Azevedo, S.F.: The simplex-simulated annealing approach to continuous non-linear optimization. *Journal of Computers and Chemical Engineering*, Vol. 20 (1996). 1065-1080.

Determining Regularization Parameters for Derivative Free Neural Learning

Ranadhir Ghosh, Moumita Ghosh, John Yearwood, and Adil Bagirov

School of Information Technology and Mathematical Sciences,
University of Ballarat, PO Box 663,
Ballarat – 3353, Australia

{r.ghosh, m.ghosh, j.yearwood, a.bagirov}@ballarat.edu.au

Abstract. Derivative free optimization methods have recently gained a lot of attractions for neural learning. The curse of dimensionality for the neural learning problem makes local optimization methods very attractive; however the error surface contains many local minima. Discrete gradient method is a special case of derivative free methods based on bundle methods and has the ability to jump over many local minima. There are two types of problems that are associated with this when local optimization methods are used for neural learning. The first type of problems is initial sensitivity dependence problem – that is commonly solved by using a hybrid model. Our early research has shown that discrete gradient method combining with other global methods such as evolutionary algorithm makes them even more attractive. These types of hybrid models have been studied by other researchers also. Another less mentioned problem is the problem of large weight values for the synaptic connections of the network. Large synaptic weight values often lead to the problem of paralysis and convergence problem especially when a hybrid model is used for fine tuning the learning task. In this paper we study and analyse the effect of different regularization parameters for our objective function to restrict the weight values without compromising the classification accuracy.

1 Introduction

Artificial neural networks (ANN) are the interconnection of basic units called artificial neurons. The performance of an ANN depends on both the weights and the transfer function. Most common form of transfer function used in literature is sigmoidal which makes the ANN model nonlinear. Using non-linear transfer function such as sigmoidal in the hidden layer provides the power of non-linearity to the network. For classification task, a choice between a linear or non-linear transfer functions exist for the output layer. Least square based methods such as householder transformation etc for finding the output layer weights often produce very large weights. This is the case especially when non-linear transfer function is used. Using linear transformations on the other hand produce less weight values for the output layer however at the expense of an unaffordable classification accuracy decrease. Large weights decrease the performance of ANN by affecting its Generalization ability, also causes paralysis during learning.

Another problem that causes due to large weight is known as network paralysis. When the weights of the neural network become very high, no change occurs in the weight updating process. So the Root mean square error does not get change. Hence the network never learns further. Often the network is required to be retrained for many purposes. Even in the simplest case using a hybrid model is sometimes a necessity for fine tuning the learning task. Adaptation is another big issue quite often faced by many learning algorithms. Leaving large synaptic weights makes retraining extremely difficult and it requires a very long training time also.

One way to avoid large weight is to apply linear transfer function in output layer. It decreases the weight in output layer but generates poor classification accuracy.

The best way to avoid the problems is Regularization. Regularization is not a new term in the ANN community [22 – 27]. It is quite often used when least square based methods or ridge regression techniques are used for finding the weights in output layer. However the term regularization is not very common for multi-layered perceptron (MLP) as it is for radial basis function (RBF) network. This can be justified by the fact that least square based techniques are not used often for MLP. Regularization increases the generalization ability and avoid overfitting. There exists several way to regularize the weights. Regularization adds a penalty term to the error function. The usual penalty is the sum of squared weights times a decay constant. This process tends to minimize the large coefficients. The generalization ability of the network can depend crucially on the decay constant. At the very least, we need two different decay constants for input to hidden layer weight, and hidden to output layer weights. Choosing the decay constant is critical issue. One way to calculate the decay constant is to iteratively update the decay constant during training. Adjusting all these decay constants to produce the best estimated generalization error often requires vast amounts of computation.

In this research we apply a local search method to find the decay constant. The derivative for local search method simultaneously minimize the classification error and estimates the decay constants which bounds the weights within certain limit.

2 The Optimization Problem

If we consider a network with differentiable activation functions, then the activation functions of the output units become differentiable functions of both the input variables and of the weights and biases. If we define an error function (E), such as the sum of squares function, which is a differentiable function of the network outputs, then this error function is itself a differentiable function of the weights. We can therefore evaluate the derivatives of the error with respect to the weights, and these derivatives can then be used to find weight values, which minimize the error function, by using a variety of learning algorithms such as: backpropagation (BP), conjugate gradient, quazi-Newton and Levenberg-Marquardt (LM) methods [1]. Viewed from the mathematical programming perspective [2, 3], supervised batch training of a neural network is a classical non-linear optimisation problem: find the minimum of the error function given some set of training data. Traditionally this is accomplished by a suitable local descent technique, such as backpropagation. The independent variables are

the weights \mathbf{w} , and the objective function is usually the sum of squared errors (although other measures of error are also used). It is formulated mathematically as

$$\min E_1(w_o, w_k) = \sum (f(w_o^T z_k) - y_k)^2, \text{ where } z_k = f(w_k^T x_k)$$

Here f denotes the transfer function (in this work the transfer function was the standard sigmoid $f(t) = (1 + e^{-t})^{-1}$), \mathbf{w}_o denote output weights, \mathbf{w}_h denote hidden layer weights, x_k are the input training data, y_k is the desired output and z_k denote the activations of hidden neurons. This problem is a global optimisation problem with many local minima.

We can consider a second error function called the absolute error function which is the summation of the absolute value of the error between the actual output and the desired output. It is mathematically formulated as

$$\min E_0(w_o, w_k) = \sum |f(w_o^T z_k) - y_k|, \text{ where } z_k = f(w_k^T x_k).$$

Despite its popularity, backpropagation has been widely criticized for its inefficiency [4, 5], and more advanced minimization techniques have been tried. Gradient based or gradient descent learning is named because of the learning characteristic of the algorithm, which uses gradient information of the error surface. Minimizing error with gradient descent is the least sophisticated but nevertheless in many cases a sufficient method. Typical response surfaces often possess local minima. Optimization techniques based on gradient descent may stagnate at these potentially sub-optimal solutions, rendering the network incapable of sufficient performance. Newton and quasi-Newton methods may also fall prey to such entrapment.

Research indicates that empirical MLP error surfaces have an extreme ratio of saddle points. The results and experience of research into the properties of the error surface have identified an important feature of MLP error surfaces, which has implications for successful training of the net. The presence of relatively steep and flat regions is a fundamental feature of the error surface. Because of the complexity of the surface it is sometimes very hard and costly to compute the derivative. This is also known as the ill conditioning of the error surface. Algorithms that do not use gradient information directly will be affected implicitly through their reliance on the values of the error function. Algorithms such as Quasi-Newton (QN) and Levenberg-Marquardt [4], which use second order information, may not converge much faster than gradient methods in such a situation, and due to their increased computation effort may actually result in slower execution times.

All these techniques converge to the closest local minimum of the error function, which is very unlikely to be the global one. As a consequence, the network trained with a local algorithm may exhibit marginal performance. In this connection, the primitive backpropagation may result in a better solution than more sophisticated methods, because its disadvantages turn to the benefits of avoiding some shallow local minima [4]. The problem of many local minima has been widely addressed in the past [3, 6]. It was shown that training even a simple perceptron with non-linear transfer function may result in multiple minima [7].

Many tricks have been invented for avoiding this problem, such as restarting with a new random set of weights, training with noisy exemplars, and perturbing the weights when they appear to prematurely converge. While these methods may lead to improved solutions, there is no guarantee that such minima will not also be only locally optimal. Further, the same suboptimal solution may be rediscovered, leading to fruitless oscillatory training behaviour. Stochastic techniques offer an alternative to conventional gradient methods. The new stochastic optimisation algorithms significantly outperform the local methods, yet they do not provide any guarantee that their solution is indeed the global minimum. What is more, the number of local minima of the error function grows exponentially with the number of neurons, and the likelihood that these stochastic methods will find the global minimum is not that high. Deterministic global optimisation techniques could be found in [8, 9, 10]. They are based on more or less systematic exploration of the search space, and involve some assumptions about the class of the error function, such as Lipschitz properties. With a suitable choice of neuron transfer functions, these properties are satisfied. The biggest problem of deterministic techniques is their computational complexity, which grows exponentially with the number of variables (weights). Hence they are applicable only to small dimensional problems. On the other hand, it is in the systems with few neurons where global optimisation techniques are most needed. Indeed one of the goals of using global optimisation in ANN training is to reduce the number of neurons without sacrificing the performance, and this has been achieved in many cases [6]. The remedies include starting local descent from several random points, using tabu search, simulated annealing and genetic algorithms.

As a stochastic process simulated annealing can serve to generate weight and bias sets [11]. The popularity of tabu search has grown significantly in the past few years as a global search technique. Glover initially introduced (1986) and later developed Tabu search [12] into a general framework. Independently, Hansen (1987) proposed the Steepest Ascent, Mildest Descent (SAMD) algorithm [13] that uses similar ideas. Tabu search can be thought of as an iterative descent method. The use of evolutionary based algorithms for training neural networks has recently begun to receive a considerable amount of attention. Although the origins of evolutionary computing can be traced back to the late 1950's, the field remained relatively unknown to the broader scientific community over the last few decades. The fundamental work of Holland, Rechenberg, Schwefel and Fogel served to slowly change this picture during the 1970s [14]. Much of the research however has focused on the training of feed forward networks [Fogel, Fogel, and Porto, 1990; Whitley, Starkweather, and Bogart, 1990] [15] [16] [17]. Just as neurobiology is the inspiration for artificial neural networks, genetics and natural selection are the inspiration of the genetic algorithm (GA). It was developed by John Holland [18]. They are based on a Darwinian type, survival of the fittest strategy. An advantage of using GAs for training neural networks is that they may be used for network with arbitrary topologies.

3 Regularization

We add the decay constant and the weights as a penalty term in the error function. As large weights in the output layer can produce outputs that are far beyond the range of

the data hence it is more important to control the output layer weights. We use a linear penalty term for the hidden layer weights and nonlinear penalty term for output layer weights. If we consider the sum of square error function as

$$\min E_1(w_o, w_k) = \sum (f(w_o^T z_k) - y_k)^2, \text{ where } z_k = f(w_k^T x_k), \text{ the regular-}$$

ized error function will be as follows

$$E_{reg}(w_o, w_k) = E_1(w_o, w_k) + w_k \lambda_k + w_o^2 \lambda_o$$

where λ_k is the matrix of all hidden layer decay constant, and λ_o is the matrix of all output layer decay constant. The actual objective function becomes

$$\min E_{reg}(w_o, w_k) = \min \left(\sum (f(w_o^T z_k) - y_k)^2 + w_k \lambda_k + w_o^2 \lambda_o \right), \text{ where } z_k = f(w_k^T x_k)$$

If we consider the absolute error function as

$$\min E_0(w_o, w_k) = \sum (|f(w_o^T z_k) - y_k|), \text{ where } z_k = f(w_k^T x_k)$$

the regularized error function will be as follows

$$E_{reg}(w_o, w_k) = E_0(w_o, w_k) + w_k \lambda_k + w_o^2 \lambda_o, \text{ where } \lambda_k \text{ is the matrix of all hid}$$

den layer decay constant, and λ_o is the matrix of all output layer decay constant.

The actual objective function becomes

$$\min E_{reg}(w_o, w_k) = \min \left(\sum (|f(w_o^T z_k) - y_k|) + w_k \lambda_k + w_o^2 \lambda_o \right), \text{ where } z_k = f(w_k^T x_k).$$

4 Method

In this section we will give a brief description of the discrete gradient method. The full description of this method can be found in [19]. The discrete gradient method can be considered as a version of the bundle method [22] when sub-gradients are replaced by their approximations - discrete gradients

Let f be a locally Lipschitz continuous function defined on R^n . A function f is locally Lipschitz continuous on R^n if in any open bounded subset $S \subset R^n$ there exists a constant $L > 0$ such that

$$f(x) - f(y) \leq L \|x - y\|, \quad \forall x, y \in S. \quad (1)$$

The locally Lipschitz function f is differentiable almost everywhere and one can define for it a set of generalized gradients or a Clarke sub-differential [93], by

$$\partial f(x) = \text{co} \left\{ v \in R^n : \exists (x^k \in D(f), x^k \rightarrow x, k \rightarrow \infty) : v = \lim_{k \rightarrow +\infty} \nabla f(x^k) \right\}$$

Here $D(f)$ denotes the set where f is differentiable, co denotes the convex hull of a set and $\nabla f(x)$ stands for a gradient of the function f at a point $x \in R^n$.

Let

$$\begin{aligned} S_1 &= \{g \in R_n : \|g\| = 1\} \\ G &= \{e \in R^n : e = (e_1, e_2, \dots, e_n), |e_j| = 1, j = 1, \dots, n\} \\ P &= \{z(\lambda) : z(\lambda) \in R^1, z(\lambda) > 0, \lambda > 0, \lambda^{-1}z(\lambda) \rightarrow 0, \lambda \rightarrow 0\} \\ I(g, \alpha) &= \{i \in \{1, \dots, n\} : |g_i| \geq \alpha\} \end{aligned}$$

where $\alpha \in \left(0, n^{-\frac{1}{2}}\right)$ is a fixed number. Here S_1 is the unit sphere, G is a set of vertices of the unit hyper cube in R^n and P is a set of univariate positive infinitesimal functions.

We define operators $H_i^j : R^n \rightarrow R^n$ for $i = 1, \dots, n, j = 0, \dots, n$ by the formula

$$H_i^j g = \begin{cases} (g_1, \dots, g_j, 0, \dots, 0) & \text{if } j < i, \\ (g_1, \dots, g_{i-1}, 0, g_{i+1}, \dots, g_j, 0, \dots, 0) & \text{if } j \geq i \end{cases}$$

We can see that $H_i^0 g = 0 \in R^n$ for all $i = 0, \dots, n$. Let $e(\beta) = (\beta e_1, \beta^2 e_2, \dots, \beta^n e_n)$, $\beta \in (0, 1]$. For $x \in R^n$ we will consider vectors

$$x_i^j(g) \equiv x_i^j(g, e, z, \lambda, \beta) = x + \lambda g - z(\lambda) H_i^j e(\beta)$$

Definition 1. The discrete gradient of the function f at the point $x \in R^n$ is the vector

$$\Gamma^i(g, e, z, \lambda, \beta) = (\Gamma_1^i, \Gamma_2^i, \dots, \Gamma_n^i) \in R^n, g \in S_1, i \in I(g, \alpha)$$

with the following coordinates:

$$\Gamma_j^i = [z(\lambda) e_j(\beta)]^{-1} [f(x_i^{j-1}(g)) - f(x_i^j(g))], j = 1, \dots, n, j \neq i$$

$$\Gamma_i^i = (\lambda g_i)^{-1} \left[f(x_i^n(g_i)) - f(x) - \sum_{j=1, j \neq i}^n \Gamma_j^i (\lambda g_j - z(\lambda) e_j(\beta)) \right]$$

Remark1: From the definition of the discrete gradient we can see that it is defined with respect to a given direction $g \in S_1$ and in order to calculate the discrete gradient we use step $\lambda > 0$ along this direction. The $n - 1$ coordinates of the discrete gradient are defined as finite difference estimates to a gradient in some neighborhood of the point $x + \lambda g$. The i th coordinate of the discrete gradient is defined so that to approximate a sub-gradient of the function f . Thus the discrete gradient contains some information about the behavior of the function f in some region around the point x .

5 Experimental Result

All experiments were conducted for 5 different datasets (Austral, Breast Cancer, Cleveland Heart Disease, Diabetes and Liver) taken from UCI ML repository. The details of these datasets can be obtained from the UCI website. All results are given for weight determination using the discrete gradient method (DG) with two different error functions, the absolute error function and the sum of squared error function. Ten fold cross validation is used with 20 % of each dataset being withheld for testing. Each experiment is conducted for a range of hidden neurons (2-8).

The following table (Table 1) shows the classification accuracy of the ANN as a percentage, CPU time in seconds and the corresponding initial weight range for the discrete gradient method with the absolute error function (Error function 0). B stands for the weight range before applying the regularization factor, and A stands for the weight range after applying the regularization factor. The average classification accuracies were same before and after applying the proposed regularization factor. But our preliminary results have shown that the existing regularization factors produce less synaptic weight values at the expense of classification accuracy.

Table 1. Results for all data sets for discrete gradient method with error function 0

Dataset	#HN	Classification Accuracy (%)	CPU Time (s)	Weight range Hidden Layer				Weight range Output Layer			
				Min		Max		Min		Max	
				B	A	B	A	B	A	B	A
Austral	3	87.8	38.65	-50	-2.5	50	1.5	-30	-5	30	10
Breast Cancer	4	100	74.13	-50	-30	50	20	-50	-8	30	15
Cleveland	2	83.3	16.13	-50	-20	50	10	-50	-5	30	25
Diabetes	2	74	14.25	-50	-20	50	15	-50	-15	30	20
Liver	3	80	12.13	-50	-10	50	20	-50	-10	30	14

The following table (Table 2) shows the classification accuracy as a percentage, CPU time in seconds and the corresponding initial weight range for the discrete gradient method with sum of squares error function (Error function 1). B stands for the weight range before applying the regularization factor, and A stands for the weight range after applying the regularization factor.

Table 2. Results for all data sets for discrete gradient method with error function 1

Dataset	#IN	Classification Accuracy (%)	CPU Time (s)	Weight range Hidden Layer				Weight range Output Layer			
				Min		Max		Min		Max	
				B	A	B	A	B	A	B	A
Austral	2	86.7	34.54	-50	-10	50	10	-30	-15	30	10
Breast Cancer	2	100	24.52	-50	-10	50	20	-50	-10	30	10
Cleveland	2	80	18.5	-50	-20	50	20	-50	-5	30	20
Diabetes	4	76.5	64.12	-50	-30	50	10	-50	-10	30	15
Liver	3	86.7	16.13	-50	-10	50	20	-50	-10	30	14

6 Conclusion and Further Research

In this paper we have proposed a new weight regularization technique for MLP learning using a derivative free optimization method without loosing any classification accuracy. Less weight values increases the generalization ability and solve the problem of paralysis. Thus it helps to retrain the network to increase its adaptability and also fine tuning the learning task by applying further a hybrid model. In future we will modify our existing discrete gradient method to incorporate the constraints as a secondary objective function.

References

- [1] Bishop, C. M.: Neural Networks for Pattern Recognition, Oxford Press, (1995).
- [2] Mangasarian, O. L.: Mathematical programming in neural networks, ORSA Journal on Computing, Vol. 5 (1993), 349--360.
- [3] Zhang, X. M., and Chen, Y. Q.: Ray-guided global optimization method for training neural networks, Neurocomputing, Vol. 30 (2000), 333-337.
- [4] Masters, T.: Practical neural network recipes in C++, Academic Press, Boston, (1993).
- [5] Masters, T.: Advanced algorithms for neural networks : a C++ sourcebook, Wiley, New York, (1995).
- [6] Duch, W., and Korczak, J.: Optimization and global minimization methods suitable for neural networks, Neural computing surveys (1999).
- [7] Coetzee, F. M., and Stonick, V. L.: On the uniqueness of weights in singlelayer perceptrons, IEEE Transactions on Neural Networks, Vol. 7 (1996), 318(8).
- [8] Horst, R. and Pardalos, P. M.: Handbook of global optimization, Kluwer Academic Publishers, Dordrecht ; Boston, (1995).
- [9] Pinter, J.: Global optimization in action : continuous and Lipschitz optimization--algorithms, implementations, and applications, Kluwer Academic Publishers, Dordrecht ; Boston, (1996).
- [10] Torn, A. and Zhitlinskas, A.: Global optimization, Springer-Verlag, Berlin ; New York, (1989).
- [11] Porto, V.W., Fogel, D. B., Fogel, L. J.: Alternative Neural Network training algorithm, Intelligent system, June (1995), Vol. 10, no. 3, 16 – 22.
- [12] Glover, F., Future path for integer Programming and Links to Artificial Intelligence”, Computer Operations Research, Vol. 13, 533 – 549.

- [13] Hansen, P., and Jaumard, B., Algorithms for the Maximum satisfiability problem, RUTCOR Research Report, 43 – 87, Rutgers University, New Brunswick, NJ.
- [14] Rechenberg, I.: Cybernetic solution path of an experimental problem, *Royal Aircraft Establishment, Library Translation* no. 1122, Farnborough, Hants, U.K, Aug, 1965.
- [15] Whitley, D., Starkweather, T., and Bogart, C.: Genetic algorithms and neural networks - optimizing connections and connectivity, *Parallel Computing*, Vol. 14, (1990), 347-361.
- [16] Montana, D. and Davis, L.: Training feed forward neural networks using genetic algorithms, *Proceedings of 11th International Joint Conference on Artificial Intelligence IJCAI-89*, Vol. 1, (1989). 762-767.
- [17] Goldberg, D. E.: *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, Reading, MA, (1989).
- [18] Holland, J. H.: *Adaptation in natural and artificial systems*, Ann Arbor, MI: The University of Michigan Press, (1975).
- [19] Bagirov, A.M.: Derivative-free methods for unconstrained nonsmooth optimization and its numerical analysis, *Investigacao Operacional*, Vol. 19, (1999), 75-93.
- [20] Bagirov, A.M.: Minimization methods for one class of nonsmooth functions and calculation of semi-equilibrium prices, *Applied Optimization*, Vol. 30: *Progress in Optimization: Contribution from Australasia*. A. Eberhard et al. (eds.), Kluwer Academic Publishers, Dordrecht, (1999), 147-175.
- [21] Bagirov, A.M.: A method for minimization of quasidifferentiable functions, *Optimization Methods and Software*, Vol. 17, No. 1, (2002), 31-60.
- [22] Hiriart-Urruty, J. B., and Lemarechal, C.: *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, New York, 1993.
- [23] Bartlett, P.L. For valid generalization, the size of the weights is more important than the size of the network. Mozer, M.C., Jordan, M.I., and Petsche, T., (eds.) *Advances in Neural Information Processing Systems 9*, Cambridge, MA: The MIT Press, (1997), 134-140.
- [24] Bishop, C.M.: *Neural Networks for Pattern Recognition*, Oxford: Oxford University Press, (1995),.
- [25] Geman, S., Bienenstock, E. and Doursat, R.: Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, Vol. 4, (1992), 1-58.
- [26] Ripley, B.D.: *Pattern Recognition and Neural Networks*, Cambridge: Cambridge University Press, (1996).
- [27] Weigend, A. S., Rumelhart, D. E., & Huberman, B. A.: Generalization by weight-elimination with application to forecasting. In: R. P. Lippmann, J. Moody, & D. S. Touretzky (eds.), *Advances in Neural Information Processing Systems 3*, San Mateo, CA: Morgan Kaufmann, (1991).

A Comprehensible SOM-Based Scoring System

Johan Huysmans¹, Bart Baesens^{1,2}, and Jan Vanthienen¹

¹ K.U.Leuven, Dept. of Applied Economic Sciences,
Naamsestraat 69, B-3000 Leuven, Belgium

² School of Management, University of Southampton,
Southampton, SO17 1BJ, United Kingdom

Abstract. The significant growth of consumer credit has resulted in a wide range of statistical and non-statistical methods for classifying applicants in ‘good’ and ‘bad’ risk categories. Traditionally, (logistic) regression used to be one of the most popular methods for this task, but recently some newer techniques like neural networks and support vector machines have shown excellent classification performance. Self-organizing maps (SOMs) have existed for decades and although they have been used in various application areas, only little research has been done to investigate their appropriateness for credit scoring. In this paper, it is shown how a trained SOM can be used for classification and how the basic SOM-algorithm can be integrated with supervised techniques like the multi-layered perceptron. Classification accuracy of the models is benchmarked with results reported previously.

1 Introduction

One of the key decisions financial institutions have to make is to decide whether or not to grant a loan to a customer. This decision basically boils down to a binary classification problem which aims at distinguishing good payers from bad payers. Until recently, this distinction was made using a judgmental approach by merely inspecting the application form details of the applicant. The credit expert then decided upon the creditworthiness of the applicant, using all possible relevant information concerning his sociodemographic status, economic conditions, and intentions. The advent of data storage technology has facilitated financial institutions ability to store all information regarding the characteristics and repayment behavior of credit applicants electronically. This has motivated the need to automate the credit granting decision by using statistical or machine learning algorithms. Numerous methods have been proposed in the literature to develop credit-risk evaluation models. These models include traditional statistical methods (e.g. logistic regression [13]), classification trees [5], neural network models [1, 4, 18] and support vector machines [2, 15]. While newer approaches, like neural networks and support vector machines, offer high predictive accuracy, it is often difficult to understand the motivation behind their classification decisions. In this paper, the appropriateness of SOMs for credit scoring is investigated.

The powerful visualization possibilities of this neural network model offer a significant advantage for understanding its decision process. However, the training process of the SOM is unsupervised and initially the predictive power lies therefore slightly below classification accuracy of several supervised classifiers. In the rest of the paper, we investigate how the SOM can be integrated with supervised classifiers. Two distinct approaches are adopted. In the first approach, the classification accuracy of individual neurons is improved through the training of a separate supervised classifier for each of these neurons. The second approach is similar to a stacking model. The output of a supervised classifier is used as input to the SOM. Both models are tested on two data sets obtained from a major Benelux financial institution and benchmarked with the results of other classifiers reported in [2].

2 Self Organizing Maps

SOMs were introduced in 1982 by Teuvo Kohonen [10] and have been used in a wide array of applications like the visualization of high-dimensional data [16], clustering of text documents [8], identification of fraudulent insurance claims [3] and many others. An extensive overview of successful applications can be found in [11] and [6]. A SOM is a feedforward neural network consisting of two layers. The neurons from the output layer are usually ordered in a low-dimensional grid. Each unit in the input layer is connected to all neurons in the output layer. Weights are attached to each of these connections. This is similar to a weight vector, with the dimensionality of the input space, being associated with each output neuron. When a training vector \mathbf{x} is presented, the weight vector of each neuron c is compared with \mathbf{x} . One commonly opts for the euclidian distance between both vectors as the distance measure. The neuron that lies closest to \mathbf{x} is called the ‘winner’ or the Best Matching Unit (BMU). The weight vector of the BMU and its neighbors in the grid are adapted with the following learning rule:

$$\mathbf{w}_c = \mathbf{w}_c + \eta(t)\Lambda_{winner,c}(t)(\mathbf{x} - \mathbf{w}_c) \quad (1)$$

In this expression $\eta(t)$ represents the learning rate that decreases during training. $\Lambda_{winner,c}(t)$ is the so-called neighborhood function that decreases when the distance in the grid between neuron c and the winner unit becomes larger. Often a gaussian function centered around the winner unit is used as the neighborhood function with a decreasing radius during training. The decreasing learning rate and radius of the neighborhood function result in a stable map that does not change substantially after a certain amount of training.

From the learning rule, it can be seen that the neurons will move towards the input vector and that the magnitude of the update is determined by the neighborhood function. Because units that are close to each other in the grid, will receive similar updates, the weights of these neurons will resemble each other and the neurons will be activated by similar input patterns. The winner units for similar input vectors are mostly close to each other and self-organizing maps are therefore often called topology-preserving maps.

3 Related Research

In [12], a model based on self-organizing maps is used to predict corporate bankruptcy. A data set containing 129 observations and 4 variables was divided into a training and a test data set with the proportion between bankrupt and solvent companies being almost equal. A 12 by 12 map was trained and divided into a zone of bankruptcy and a zone of solvency. This division of the map was obtained by labelling each neuron with the label of the most similar training example. Unseen test observations were classified by calculating the distance between the neurons of the map and the observations. If the most-active neurons were in the solvent zone, the observation was classified as good. It is concluded that the percentage correctly classified observations is comparable with the accuracy of a linear discriminant analysis and several multi-layered perceptrons. The author's conclusion is promising for the SOM: the flexibility of this neural model to combine with and to adapt to other structures, whether neural or otherwise, augurs a bright future for this type of model.

In [9], several SOM-based models for predicting bankruptcies are evaluated. The first of the models, SOM-1, is very similar to the model described above, but instead of assigning each neuron the label of the most similar observation, a voting scheme is used. For each neuron of the map, the probability of bankruptcy is estimated as the number of bankrupt companies projected onto that node divided by the total number of companies projected on that neuron. A second, more complex model was also proposed (SOM-2). It consists of a small variation to the Basic SOM-algorithm as explained above. Each input vector consists of two types of variables: the financial indicators and the bankruptcy indicators. Only the financial indicators are used when searching which unit is the BMU. Afterwards, the weights are updated with the traditional learning rule from equation 1. These weight updates are not only made for the financial indicators but also for the bankruptcy indicators. The weight of the bankruptcy indicator after training is used as an estimate for the conditional probability of bankruptcy given the neuron. Compared to other classifiers, like LDA and LVQ, SOM-1 was clearly outperformed. SOM-2 performed much better and more importantly: its classification accuracy was quite insensitive to the map grid size.

4 Description and Preprocessing of the Data

For this application, two different data sets were at our disposal. The characteristics of these data sets are summarized in Table 1. The same data sets are described in detail in a benchmarking study of different classification algorithms [2]. In this benchmarking study, two thirds of the data were used for training and one third as test set. The same training and test sets will be used in this paper. Additional measures like sensitivity and specificity for these classifiers are also given in Table 1. Sensitivity measures the number of good risks that are correctly identified while specificity measures the number of bad risks that are correctly classified.

Table 1. Description of the Datasets

name	Bene1	Bene2
number of obs.	3123	7190
number of variables	27	27
good/bad	67:33	70:30
best classifier	RBF LS-SVM(73.1%)	MLP(75.1%)
sens/spec	83.9%/52.6%	86.7%/48.1%

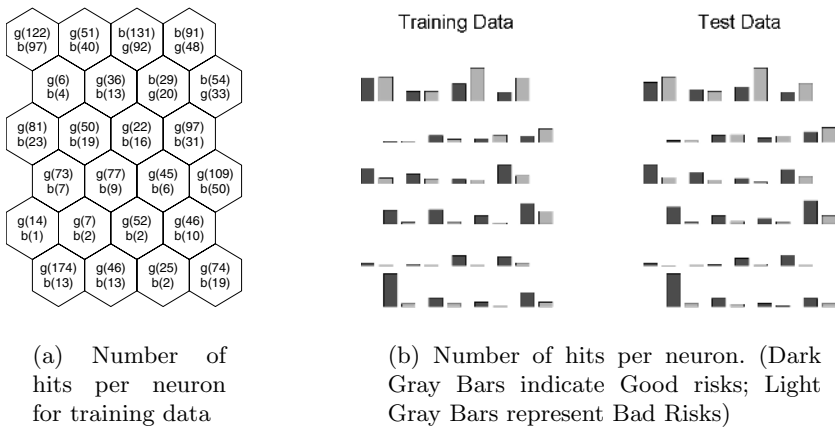
Both data sets contain several categorical variables, like *goal of the loan* and *residential status*. A weights of evidence encoding [14] was performed to transform them into numerical variables. After performing the weights of evidence encoding for the categorical variables, an additional normalization was done for all variables.

5 Exploratory Data Analysis

5.1 Visualization of the SOM

SOMs have mainly been used for exploratory data analysis and clustering. In this section, the basic SOM-algorithm will be applied to the Bene1 data set. A map of 6 by 4 neurons is used because it is small enough to be conveniently visualized. All analyzes are performed with the SOM-toolbox for Matlab [7].

To examine if ‘good’ and ‘bad’ risk observations are projected onto different units, we can calculate for each observation the winner neuron. For the Bene1 data set, this results in Figure 1(a). In each neuron, the number of good and bad risk observations that were projected onto that neuron, are given. For example,

**Fig. 1.** Number of hits per neuron

the upper-left neuron was the BMU for 219 training observations, from which 122 were good and 97 were bad. The same information is also given in the left part of Figure 1(b). In this figure, the size of the bar indicates the number of good and bad observations projected onto each neuron. Notice that the scale of the bars is different for ‘good’ and ‘bad’ risk categories. The right part of Figure 1(b) contains the same information, but this time for the unseen test data. From the graphs, it can be noticed that bad risks tend to be projected onto the neurons in the upper half of the grid, but that the SOM is not able to achieve a clear separation. This corresponds with the results from [2], even powerful techniques like Support Vector Machines are not able to obtain a very high degree of accuracy on the Bene1 data set.

6 Classification

The SOM we created can also be used for classification. In [12], a SOM is created and each neuron is assigned the label of the closest training observation. Predictions for the test data are based on the label of their BMU. Using the same labelling on our map of the Bene1 data, results in 4 nodes that are assigned the bad status and 20 nodes a good status. The labelling is shown in Figure 2(a). It can be seen that most of the nodes labelled ‘bad’ are situated in the lower part of the map. From Figure 1(b) however, we know that most bad risk observations are projected on the upper half of the map. The accuracy, specificity and sensitivity of this classification method are therefore rather low (respectively 58%, 22% and 76%). Changes in grid size do not considerably alter these results. For the Bene2 data set, with a grid of 6 by 4, accuracy, specificity and sensitivity are respectively 66%, 14% and 88%. These numbers are considerably below the performance of several supervised classifiers reported in [2]. Instead of using only the closest training observation for labelling each neuron, more sophisticated techniques, like k-nearest neighbor, might prove useful.

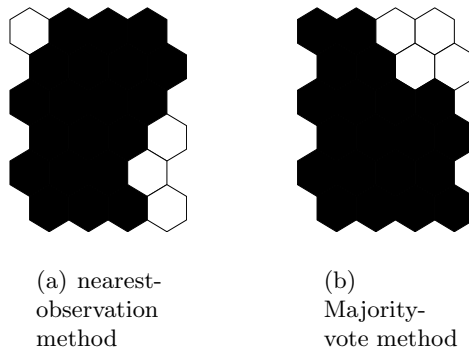


Fig. 2. Classification of neurons (White: nodes assigned Bad status, Black: nodes assigned Good Status)

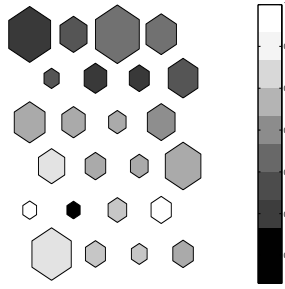


Fig. 3. Accuracy in each node for Majority-vote method (Bene1 Test Data)

A second way of labelling the neurons was proposed in [9]: each node receives the label of the class from which most training observations are projected on the node. This method will always result in a greater classification accuracy on the training data. Figure 2(b) shows this labelling for the Bene1 data with a 6 by 4 grid. The accuracy, specificity and sensitivity of this map are respectively 71.0%, 44.57% and 84.9%. For the Bene2 data, with the same map size, classification performance was 69.7% because the model assigns the ‘good’ label to all-but-one neuron and will therefore mainly just predict the majority class.

However, it is more interesting to identify the neurons of the map that are responsible for most misclassifications. Figure 3 gives an overview of the classification accuracy in each node for the Bene1 data. Dark nodes are neurons with low classification accuracy. The size of the neurons is an indicator of the number of observations for which that neuron is the BMU. We can see that some neurons are the BMU for lots of observations, while others are the BMU for only a few examples. The presence of large and dark neurons in Figure 3 will indicate a bad classification accuracy of the map. For the Bene1 data set, it can be seen that the lower part of the map has a good classification accuracy. The upper half of the grid shows worse accuracy. For some nodes, the accuracy is below 50%. Fortunately, only few observations are projected onto these nodes. From the figure, we conclude that many observations are projected onto the first neuron of the top row and that not all of these observations belong to the category ‘good’, because classification accuracy is low. In the following section, a more detailed model is elaborated. Observations that are projected onto neurons with low accuracy, will not receive the standard labelling of these nodes, but will instead be classified by independent models. If a SOM is used as new model, a hierarchy of SOMs for prediction is obtained.

7 Integration of SOMs with Other Classification Techniques

In the previous sections, we have shown that SOMs offer excellent visualization possibilities that leads to a clear understanding of the decisions made by

these models. But due to their unsupervised nature, SOMs seem not able to obtain the degree of accuracy achievable by several supervised techniques, like multi-layer perceptrons or support vector machines. In this part, the classification performance of the SOMs will be improved by integrating them with these supervised algorithms. There are two possible approaches for obtaining this integration. One possibility is to first train a SOM and then use the other classification techniques to improve the decisions made by the individual neurons of this SOM. A second possibility is to use the predictions of the supervised classifiers as inputs of the SOM. These two approaches are now discussed in more detail.

7.1 Improving the Accuracy of Neurons with Bad Predictive Power

From Figure 3, we can observe that not all neurons achieve the same level of accuracy when predicting the risk category of an applicant. The lack of accuracy of the predictions made by the neurons in the top rows is compensated by the almost perfect predictions in the lower half of the map. A two-layered approach is suggested in this section. For neurons that achieve almost perfect accuracy on the training data when using one of the models from the previous section, nothing changes. All the observations projected on one of these neurons, will therefore receive the same label. There are only changes for neurons whose level of accuracy on the training data lies below a user-specified threshold. For each of these neurons, we build a classifier based on the training examples projected on that neuron. In our experiments, we used feedforward neural networks as classifiers for each of the neurons, but there is no necessity for the classifiers being of the same type. The user-specified accuracy threshold was fixed at 58% for the Benel data set with a 6 by 4 map. This value has been estimated by a trial-and-error procedure. A threshold that is set too low will give no improvement over the above mentioned classifiers because no new models will be estimated. The opposite, a very high threshold, will result in too many new classifiers to be trained. With a threshold of 58%, three models will be trained: two for the first two neurons of the top row and one for the third neuron of the third row. We tested with several different values for the number of hidden neurons in the neural networks. The simplest case, with only one hidden neuron delivered best results with an accuracy on the test set of 71.3% averaged over 100 independent trials. This is almost no improvement over the majority vote classifier that showed an accuracy of 71.0%. It seems that the increase in accuracy of the 3 newly trained classifiers is only marginal. For some neurons, a decrease in the percentage correctly classified can even be noted. It seems extremely difficult to separate the applicants that are projected on these neurons. A possible improvement might result from requesting additional information if an applicant is projected on one of the low-accuracy neurons and then training the feedforward neural networks with this additional information. For applicants that are projected on one of the other neurons, requesting this additional informa-

tion is not necessary. The results for Bene2 are similar. A threshold of 65% results in 9 additional classifiers to be trained of which most are situated in the upper half of the map. The accuracy, averaged over 100 independent trials, improves to 71.9% compared with the original performance of the majority-vote method of 69.7%. Specificity and sensitivity are respectively 26.8% and 91.1%.

7.2 Stacking Model

A stacking model [17] consists of one meta-learner that combines the results of several base learners to make its predictions. In this section, a SOM will be used as the meta-learner. The main difference with the previous section is that the classifiers are trained before training the SOM and not afterwards. The classifiers also learn from all available training observations and not from a small subpart of it.

In our experiments, we start with only one base learner, a multi-layer perceptron with 2 hidden neurons, which achieves an average classification accuracy of 72.5% on the Bene1 data set (75.1% on Bene2). The input of the meta-learner, the SOM, consists of the training data augmented with the output of this MLP. A small variation to the above described basic SOM-algorithm is used. Instead of finding the BMU by calculating the euclidian distance between each neuron and the sample observation, a weighting factor is introduced for each variable. Heavily weighted variables, in our case the output from the MLP, will contribute more during formation of the map. The distance measure, with n the number of variables, can be written as:

$$\| \mathbf{x} - \mathbf{w}_c \| = \sum_{i=1}^n weight_i | x_i - w_{c,i} |^2 \tag{2}$$

The update rule from equation 1 does not change. So introducing the weights only affects finding the BMU's of the SOM [7].

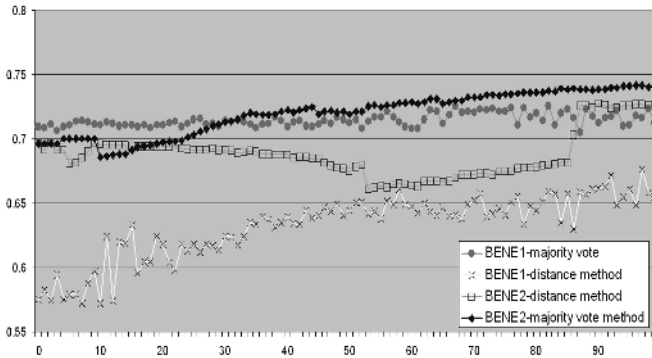


Fig. 4. Stacking Model: Accuracy in function of weighting factor

Table 2. Overview of Classification Results (accuracy, specificity and sensitivity)

Classifier	Bene1	Bene2
NTO	57.5 / 22.3 / 76.1	66.1 / 14.5 / 88.3
MV	71.0 / 44.6 / 84.9	69.7 / 00.6 / 99.34
IA	71.3 / 52.6 / 81.5	71.9 / 26.8 / 91.1
SM	Fig. 4	Fig. 4
C4.5	68.9 / 52.6 / 77.4	69.8 / 43.0 / 81.3

NTO=‘Nearest Training Observation’-method, MV= ‘Majority vote’-method, IA= Improving Accuracy of Neurons with bad Predictive Power, SM= Stacking Model

For the experimental study, all the weighting factors of the original variables were fixed at a value of one while the weighting factor of the MLP-input was varied between 1 and 100. Classifications were made by both methods discussed above: the majority-vote method and the nearest-distance method. Figure 4 gives an overview of the classification accuracy for each method and for both data sets with a grid size of 6 by 4.

It can be seen that performance of the nearest-distance method is always below the performance of the majority-vote method. Second, we conclude that the weighting factor of the MLP plays a crucial role in the classification performance of the integrated SOM. In general, the larger the weighting factor is, the more the output of the integrated SOM resembles the output of the MLP. There is however a large amount of variance present in the results. A small change in weighting factor can significantly change the performance observed.

In theory, the stacking model can be used in combination with the previous method of integration, but the degree of complexity of the resulting model is high and the advantage of the SOM’s explanatory power is lost. This approach was therefore not analyzed in greater detail.

8 Conclusion

In this paper, the appropriateness of self organizing maps for credit scoring has been investigated. It can be concluded that integration of a SOM with a supervised classifier is feasible and that the percentage correctly classified applicants of these integrated networks is higher than what can be obtained by employing solely a SOM. The first method, which trains additional classifiers for neurons with bad predictive power withstands competition of other white-box techniques like C4.5. Several topics are still open for future research. For instance, we did not investigate in detail what the influence of the map size is on the results. A combination of SOMs with several different types of supervised classifiers was also not tested. Comparison of the component planes of these different classifiers might visually show where the predictions of the models agree and where they disagree.

References

1. A. Atiya. Bankruptcy prediction for credit risk using neural networks: A survey and new results. *IEEE Transactions on Neural Networks*, 12(4):929–935, 2001.
2. B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen. Benchmarking state of the art classification algorithms for credit scoring. *Journal of the Operational Research Society*, 54(6):627–635, 2003.
3. P. Brockett, X. Xia, and R. Derrig. Using kohonen’s self-organizing feature map to uncover automobile bodily injury claims fraud. *International Journal of Risk and Insurance*, 65:245–274, 1998.
4. M.-C. Chen and S.-H. Huang. Credit scoring and rejected instances reassigning through evolutionary computation techniques. *Expert Systems with Applications*, 24:433–441, 2003.
5. R. Davis, D. Edelman, and A. Gammerman. Machine learning algorithms for credit scoring applications. *IMA Journal of Mathematics Applied in Business and Industry*, 4:43–51, 1992.
6. G. Deboeck and T. Kohonen. *Visual Explorations in Finance with selforganizing maps*. Springer-Verlag, 1998.
7. Helsinki University of Technology. Som toolbox for matlab, 2003.
8. T. Honkela, S. Kaski, K. Lagus, and T. Kohonen. WEBSOM—self-organizing maps of document collections. In *Proceedings of WSOM’97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6*, pages 310–315. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, 1997.
9. K. Kiviluoto. Predicting bankruptcies with the self-organizing map. *Neurocomputing*, 21:191–201, 1998.
10. T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.
11. T. Kohonen. *Self-Organising Maps*. Springer-Verlag, 1995.
12. C. Serrano-Cinca. Self organizing neural networks for financial diagnosis. *Decision Support Systems*, 17:227–238, 1996.
13. A. Steenackers and M. Goovaerts. A credit scoring model for personal loans. *Insurance: Mathematics and Economics*, 8(1):31–34, 1989.
14. L. Thomas. A survey of credit and behavioural scoring; forecasting financial risk of lending to consumers. *International Journal of Forecasting*, 16:149–172, 2000.
15. T. Van Gestel, B. Baesens, J. Garcia, and P. Van Dijke. A support vector machine approach to credit scoring. *Bank en Financierwezen*, 2:73–82, 2003.
16. J. Vesanto. Som-based data visualization methods. *Intelligent-Data-Analysis*, 3:111–26, 1999.
17. D. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1991.
18. M. Yobas and J. Crook. Credit scoring using neural and evolutionary techniques. *IMA Statistics in Finance, Journal of Mathematics Applied in Business and Industry*, 11:111–125, 2000.

The Convex Subclass Method: Combinatorial Classifier Based on a Family of Convex Sets

Ichigaku Takigawa¹, Mineichi Kudo², and Atsuyoshi Nakamura²

¹ Bioinformatics Center, Institute for Chemical Research,
Kyoto University, Gokasho, Uji,
Kyoto 611-0011, Japan
takigawa@kuicr.kyoto-u.ac.jp

² Graduate School of Information Science and Technology,
Hokkaido University, Kita 13, Nishi 8,
Kita-ku, Sapporo 060-8014, Japan
{mine, atsu}@main.ist.hokudai.ac.jp

Abstract. We propose a new nonparametric classification framework for numerical patterns, which can also be exploitable for exploratory data analysis. The key idea is approximating each class region by a family of convex geometric sets which can cover samples of the target class without containing any samples of other classes. According to this framework, we consider a combinatorial classifier based on a family of spheres, each of which is the minimum covering sphere for a subset of positive samples and does not contain any negative samples. We also present a polynomial-time exact algorithm and an incremental randomized algorithm to compute it. In addition, we discuss the soft-classification version and evaluate these algorithms by some numerical experiments.

1 Introduction

The goal of pattern classification is, given a training set as examples, to develop a *classifier* which can assign the class label to any possible patterns in the feature space and minimizes the probability of error[1, 2, 3]. We consider the classification on the feature space \mathbb{R}^d such that all patterns are described as d numerical measurements (features). Thus, an m -class classification involves partitioning the feature space into m disjoint regions corresponding to each class. Such regions should consist of points which are likely to belonging to that class.

In pattern classification, we can use only a *finite* training set although the background probability distribution is often unknown. Moreover, if we assume the i.i.d. property behind data, the training patterns must be very carefully labeled as example patterns, thus it is often a heavy task and requires high cost to obtain a good training set in general. Consequently, the size of training set is often too small to obtain enough result by classic statistical methods.

Hence we focus attention on the nonparametric classification framework motivated by Vapnik's principle[4] "When solving a given problem using a restricted

amount of information, try to avoid solving a more general problem as an intermediate step.” Even for given training samples, we will need a nonlinear discrimination in general. Thus, we introduce a decomposition of such a complicated discriminative structure of data into smaller, easy-to-handle convex pieces.

This paper proposes such a framework as a generalization of the subclass method based on rectangles by Kudo *et al.*[5]. According to that, we develop the new combinatorial classifier based on spheres.

2 The Convex Subclass Method

2.1 General Methodology

We focus attention on a geometric intuition for problems such as how data are and how the classification will be done. Any learning algorithms encode some a priori knowledge on the given problem. Actually, many conventional classifier uses, explicitly or implicitly, some kind of computational geometric structures to classify incoming patterns. For examples, SVM uses a hyperplane (or a halfspace) and Nearest neighbor method uses a Voronoi diagram. Using a hyperplane is the simplest way to distinguish two classes, but it is still unsure whether it fits tasks for more than 3 classes or not.

In our approach, we consider representing the dispersion of each class data against other classes by covering all samples of each class with some simple convex sets (such as boxes, balls, ellipsoids, halfspaces, convex hulls, or cylinders) which does not contain any samples of other classes. Each convex set $R(Z)$ is defined by a certain subset Z of positive samples¹ (Figure 1). When given such convex sets for each class, we can assign the class label to every point $x \in \mathbb{R}^d$, based on the minimum distance between the convex sets for each class and the point x .

Convex subclasses are a family of subsets of positive samples (it forms a hypergraph) constrained by all negative samples and the type of used convex set. This idea is motivated by Kudo’s subclass method [5, 6] which uses the minimum bounding box (i.e. axis-parallel rectangle) containing the subset Z as the corresponding convex set $R(Z)$. Rectangles are, however, sometimes not suitable for given classification problems because they depend on the choice of the coordinate systems, there may exist too long and thin boxes, or the resultant decision boundary is not smooth enough.

Thus, in this paper, we extend the original subclass method *et al.*[5] to more general framework and develop the method based on spheres according to it. Our method can give a combinatorial classifier which can be exploitable for exploratory data analysis of given classification problem, such as examining the difficulty (or complexity) of problem or the effectiveness of used features. Moreover, this framework can introduce relaxation for the exclusion of negative samples, and also have a potentiality for realizing a parallel computable classifier.

¹ Besides Z , we can also use all negative samples in order to define a consistent convex set $R(Z)$, but we do not mention it here.

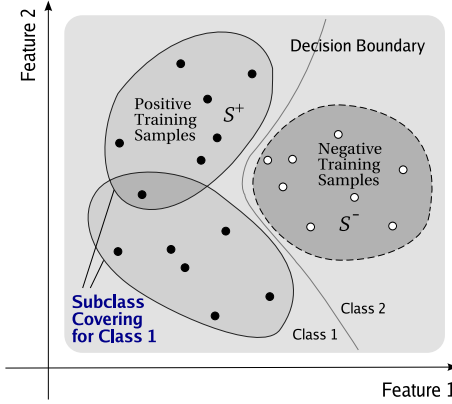


Fig. 1. The idea of the convex subclass method

2.2 Subclass Covering for Target Class

Now, we can give more formal definition of our framework. Given two finite point sets $S^+, S^- \subset \mathbb{R}^d$ as a positive set and a negative set for the target class, respectively. In this paper, although we regard $R(Z)$ as the smallest enclosing ball of a point set Z basically, any computable convex set will be available if it can be defined by only $Z \subset S^+$ (and possibly S^-).

Definition 1 (Subclass Cover). Let $R(Z) \subset \mathbb{R}^d$ be a convex set defined by a point set $Z \subset \mathbb{R}^d$. The *subclass family* \mathcal{F} of S^+ (against S^-) is a family of subsets of S^+ , which satisfies the following conditions:

- 1.1 Inclusion of positive samples: $S^+ \subset (\cup_{Z \in \mathcal{F}} R(Z))$,
- 1.2 Exclusion of negative samples: $S^- \cap (\cup_{Z \in \mathcal{F}} R(Z)) = \emptyset$,
- 1.3 Maximality of each element: for each $Z \in \mathcal{F}$,
 $\forall W \subset S^+ \setminus Z, S^- \cap R(Z \cup W) \neq \emptyset$.

We call each subset $Z \in \mathcal{F}$ a *subclass*. If 1.1 and 1.2 are satisfied, the subclass family is said to be *feasible*. We can obtain the *unique* subclass family by collecting subsets which satisfy 1.1-1.3 among all subsets.

In other words, for the union of $R(Z), Z \in \mathcal{F}$, the condition 1.1 means “it contains all positive samples”, the condition 1.2 means “it cannot contain any negative samples”, and the condition 1.3 means that for any $Z \in \mathcal{F}$, if we add any other positive samples to Z , it must violate the condition 1.2. In addition, from the condition 1.2, each $R(Z), Z \in \mathcal{F}$ cannot also contain any negative samples (Figure 2).

2.3 Weak Subclass and Relaxed Subclass

Computation of subclass is often demanding. We can use a *weak* subclass instead which is approximately maximal. This weak subclass is often sufficient for pattern classification, and it can reduce the computational cost as we see later.

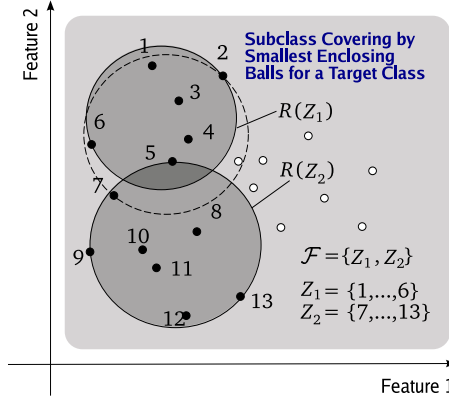


Fig. 2. An example of subclass covering by smallest enclosing balls of subsets. Maximal condition means that we cannot add any other positive samples W to Z_1 (and Z_2). In this example, $W = \{7\}$ violates exclusion of negative samples (*dashed circle*)

For this purpose, we define another condition instead of maximality. A subclass family \mathcal{F} has no elements which becomes a subset of other element. Such a family of subsets is called *Sperner family*[7] (also known as *antichain*), and hence we call this property *Sperner condition*.

Definition 2 (Weak Subclass). We consider the Sperner condition

$$1.4. \text{ Sperner Condition: } A \in \mathcal{F} \Rightarrow \forall B \in \mathcal{F} \setminus \{A\}, A \not\subseteq B,$$

as a weakened condition instead of the maximality condition 1.3 in Definition 1. We call the subclass family satisfying 1.1, 1.2 and 1.4 a *weak subclass*.

Strong subclasses always produce perfectly a consistent hypothesis, but in some applications we often need to tolerate training error in order to avoid overfitting. We can relax the condition 1.1 or 1.2. The relaxed condition can depend on the type of $R(\cdot)$ or the type of problem, and we will give later the definition of *relaxed subclass* for spheres in the subsection 3.3.

2.4 Related Previous Approach: Class Cover Problem

In this paper, we will examine the subclass method based on balls. From this viewpoint, we here refer the previously proposed framework called *class cover problem* which has a similar flavor to the subclass cover problem with balls, and discuss the difference between them.

To the best of our knowledge, the class cover problem was introduced by Cannon and Cowen[8] originally as a conference paper in 2000. Subsequently, Priebe *et al.*[9], Marchette[10], and DeVinney[11] studied and developed this framework, and proposed the graph-theoretic method called *Class Cover Catch Digraph(CCCD)* for computing it.

Let $B(c, r)$ be a ball centered at $c \in \mathbb{R}^d$ with a radius $r \in \mathbb{R}$. The class cover problem is the following problem: Suppose we draw a ball centered at each positive sample with a radius of the distance to the nearest negative sample. Then, find a *minimal* subset of positive samples, such that the corresponding balls can cover all positive samples. This problem is a generalization of the classic set cover problem (see for example [12]).

The subclass cover appears to be an unconstrained and inhomogeneous class cover neglecting that the balls are open or close, but it is not necessarily the minimal family; the class cover problem requires the minimality of a resultant cover while the subclass cover problem requires the maximality of each subclass. Another difference is that the class cover considers a subset of S^+ whereas the subclass cover considers a *family* of subsets.

The previous work mainly focused on the constrained class cover, which can give a method for prototype selection from positive samples, and can provide a prototype-based classifier[10]. Although the class cover problem of this type becomes NP-hard problem[8, 13] unfortunately, the subclass cover problem has a polynomial-time algorithm as presented later.

The property of such a multi-spheres classifier is also discussed by Adam *et al.*[13]. This study extended the classical Vapnik-Chervonenkis learning theory to the data-dependent hypothesis classes. As an example, they discussed the constrained class cover classifier, and showed some interesting properties.

In these contexts, the proposed classification method based on spheres will be also interesting.

3 The Subclass Method Based on a Family of Spheres

We will develop the subclass method based on balls. Hereafter, $R(Z)$ denotes the minimum enclosing ball for the point set Z . It should be noted that the minimum enclosing ball for a set consisting of only one point is defined as a ball centered the point with radius 0.

3.1 Algorithms for Constructing Subclass Family

Exact Algorithm. First, we show a polynomial-time exact algorithm which can enumerate all sets in the unique subclass family of target class. This is based on the simple fact that a sphere in \mathbb{R}^d can be determined by at most $d+1$ points[14]. Thus, it is always available when $R(Z)$ is defined by at most d points and the number d does not depend on the size $|Z|$.

Algorithm 1. For two given point sets S^+ and S^- , do the following:

1. Set $\mathcal{H} := \{V \subset S^+ : |V| \leq d+1\}$.
2. Remove sets which cannot exclude negative samples from \mathcal{H} .
3. Set $\mathcal{F} := \{S^+ \cap R(V) : V \in \mathcal{H}\}$ and eliminate duplication.
4. For each element in \mathcal{F} , if it becomes a subset of other element, remove it.

Roughly speaking, this algorithm first enumerates all subsets of size at most $d + 1$ which can exclude all negative samples. Then, we can obtain a subclass family \mathcal{F} as an irreducible set with respect to the Sperner condition 1.4.

Note that the condition 1.3 is satisfied. Suppose \mathcal{F} does not satisfy the maximality condition 1.3, there exists $Z \in \mathcal{F}$ such that $\exists W \subset S^+ \setminus Z, S^- \cap R(Z \cup W) = \emptyset$. Since this $Z \cup W$ can exclude all negative samples, therefore $Z \cup W \in \mathcal{F}$ contradicting our assumption for the existence of Z because $Z \subset Z \cup W$ must be removed in step 4. Hence this algorithm can enumerate all elements of the unique subclass family of S^+ .

The number of subsets of size at most $d + 1$ is polynomial with respect to the size of inputs. Assuming that the other steps requires only polynomial-time, Algorithm 1 is also polynomial-time computable. This is one of the advantages against the constrained class covers, which are NP-hard[8, 13].

Incremental Algorithm. Practically, since the computational cost of Algorithm 1 is still high, improvement of its efficiency should be required. Instead of the enumeration of unique subclasses, we consider the enumeration of elements in any of weak subclass families. In this approach, the maximality and the uniqueness of subclass family are not always satisfied, but such a subclass family is often sufficient in order to construct a pattern classifier.

Algorithm 2. For two given point sets S^+ and S^- , do the following:

1. Let D be a randomly ordered set of S^+ .
Set $C \leftarrow \emptyset$ for the set of tested points, $\mathcal{F} \leftarrow \emptyset$ for the output family, respectively.
2. Repeat the following until $D \setminus C = \emptyset$ is satisfied:
 - (a) Select randomly $x \in D \setminus C$, set $Z \leftarrow \{x\}$ and $C \leftarrow C \cup \{x\}$.
 - (b) For all $\tilde{x} \in D \setminus \{x\}$, do the following sequentially: If a point set $Z \cup \{\tilde{x}\}$ can exclude S^- then, set $Z \leftarrow Z \cup \{\tilde{x}\}$ and $C \leftarrow C \cup \{\tilde{x}\}$.
 - (c) $\mathcal{F} \leftarrow \mathcal{F} \cup Z$.
3. After eliminating duplication, for each element in \mathcal{F} , if it becomes a subset of other element, remove it.

3.2 Classification Based on Subclass Family

We now turn to the pattern classification problem. To implement the original idea described in section 2.1, we use the directed length of the minimal projection onto spheres for classifying the test samples. The directed length of projection of the point x onto sphere $B(c, r)$ is defines as

$$\tilde{d}(x, B(c, r)) := \|x - c\| - r.$$

It should be noted that if the point x is in $B(c, r)$, the value of $\tilde{d}(x, B(c, r))$ becomes negative.

For given subclass families $\mathcal{F}_1, \dots, \mathcal{F}_C$ for each class $i = 1, \dots, C$, the classification is based on

$$f(x) := \arg \min_{i=1, \dots, C} \min_{Z \in \mathcal{F}_i} \tilde{d}(x, R(Z)).$$

We can have no training error when the exclusion of negative samples are perfect. Under this quasi-distance, each subclass ball acts like prototypes for the corresponding class.

3.3 Relaxed Subclass Family for Balls

As touched in 2.3, the perfect exclusion of negative samples often yields overfitting for practical problems; Thus we often need the relaxed version of exclusion condition to tolerate training error.

In Definition 1 of subclass family, the condition 1.2 can be relaxed. As a benefit from the formalization, we can easily develop the soft-classification version of subclass method by replacing the condition 1.2 by the relaxed condition. For both Algorithm 1 and 2, the required modification is only this replacement when we check the exclusiveness of subclass. In soft-classification version, for a given parameter ξ , “ $B(c, r)$ can exclude negative samples” means

$$r = 0 \quad \text{or} \quad \sum_{x \in S^-} \max\left(0, 1 - \frac{\|x - c\|}{r}\right) \leq \xi.$$

From the definition, when $\xi = 0$, it is consistent with the perfect exclusion of negative samples (i.e. hard-classification version). In addition, we consider the second additional condition: For a given parameter δ ,

$$\delta > \frac{\# \text{ of containing negatives}}{\# \text{ of all negatives}}.$$

This additional condition is sometimes needed for avoiding excessively incorporation of negative samples to the subclass ball when we use the relaxed condition.

3.4 Computational Issues

Monotonicity of Representation. We identify $R(\cdot)$ with a function that maps any $Z \subset \mathbb{R}^d$ to $R(Z)$. We call $R(\cdot)$ a *representation*. For a given representation $R(\cdot)$ and any two point sets $U, V \subset \mathbb{R}^d$, if $U \subset V \Rightarrow R(U) \subset R(V)$ holds true, we say that the representation $R(\cdot)$ is *monotonic*.

The axis-parallel rectangles are monotonic. When the representation is monotonic, the incremental algorithm does not violate the maximality condition. However the minimum enclosing balls are non-monotonic, and thus the incremental algorithm will compute just a approximation of maximal subclasses. It should be noted that the exact algorithm can enumerate the unique subclasses in both cases.

Minimum Enclosing Ball Computation. For an implementation of Algorithm 1 or 2, the efficient method computing the minimum enclosing ball for a given point set is required. Computation of the minimum enclosing ball has a long history[14] and many algorithms have been developed. Recently, computation in higher-dimensional space or computation for large-scale problem has been

studied. Our implementation is based on the simple algorithm [15] which works efficiently for $d < 30$. For more higher-dimensional problems, we can use alternatively the computational geometric method[16] or the aggregation function method and second-order cone programming-based method[17].

4 Examples

In Figure 3, we showed the illustrative example in 2-dimensional classification problem including the result by class cover catch digraph method[9] for comparison. The results were computed by Algorithm 1 and we can see that the original idea of convex subclass method described in section 2.1 was realized well.

In order to examine the behavior for more higher dimensional data, we compared three methods: (1) the subclass method based on balls, (2) the relaxed subclass method, (3) support vector machines [4] with Gaussian kernel $K(x, y) := \exp(-\gamma\|x - y\|^2)$ and a regularization parameter C , and (4) k -nearest neighbor method. The numerical experiments were based on 10 fold cross-validation for 3 numerical datasets from UCI machine learning repository[18]: **iris** (4 features, 3 classes, 150 samples), **glass** (9 features, 6 classes, 214 samples), and **wine** (13 dimensional, 3 classes, 178 samples). The result shown in Table 1 was computed by Algorithm 2. Therefore, it depends on randomness in Algorithm 2 and the obtained subclass is not unique. But the result seems to be good enough compared with the conventional classifiers and the approximated subclasses will work well in higher-dimensional spaces. We can also see the effect of the second additional condition.

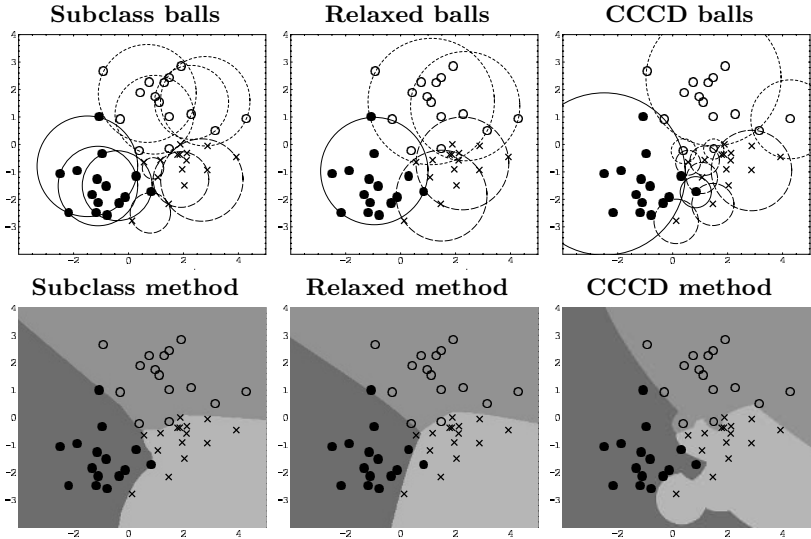


Fig. 3. Balls and decision boundaries of subclass method, relaxed subclass method with $\xi = 1$, and class cover catch digraph method[9]

Table 1. Estimated classification rate by 10-fold CV (correct number)

	Subclass	Subclass($\xi = 0.5$)		SVM($C = 1$)		SVM($C = 100$)		k -NN	
		δ		γ		γ		k	
		-	0.1	0.01	0.25	0.01	0.25	1	5
iris	96.0	90.0	94.0	88.0	96.0	96.0	94.7	96.0	96.0
(150)	(144)	(135)	(141)	(133)	(144)	(144)	(142)	(144)	(144)
glass	72.0	63.1	65.0	50.5	68.7	67.3	61.9	70.1	65.9
(214)	(154)	(135)	(139)	(108)	(147)	(144)	(136)	(150)	(141)
wine	94.9	89.9	94.9	97.8	96.6	96.1	97.2	94.9	96.1
(178)	(169)	(160)	(169)	(174)	(172)	(171)	(173)	(169)	(171)

5 Conclusion

We proposed a new nonparametric classification framework: The (convex) subclass method. According to that, we developed a combinatorial classifier based on a family of spheres, and showed a polynomial-time exact algorithm and an incremental algorithm. Additionally, the relaxed subclasses were considered and through some numerical examples we confirmed its effectiveness. Further researches will include some theoretical analysis on the dependency of randomness and the expected computational cost, developing more efficient computational methods, implementing parallel computing of subclasses, examining the subclasses based on various convex sets, and considering better relaxed conditions.

References

1. Bousquet, O., Boucheron, S., Lugosi, G.: Theory of classification: A survey of recent advances. *ESAIM Probability and Statistics*, (to appear) (2004)
2. Devroye, L., Györfi, L., Lugosi, G.: *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York (1996)
3. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification 2nd Ed.* John Wiley & Sons (2001)
4. Vapnik, V.N.: *The Nature of Statistical Learning Theory 2nd Ed.* Springer-Verlag, New York (2000)
5. Kudo, M., Yanagi, S., Shimbo, M.: Construction of class regions by a randomized algorithm: A randomized subclass method. *Pattern Recognition* **29** (1996) 581–588
6. Takigawa, I., Abe, N., Shidara, Y., Kudo, M.: The boosted/bagged subclass method. *International Journal of Computing Anticipatory Systems* **14** (2004) 311–320
7. Erdős, P., Kleitman, D.: Extremal problems among subsets of a set. *Discrete Mathematics* **8** (1974) 281–294
8. Cannon, A.H., Cowen, L.J.: Approximation algorithms for the class cover problem. *Annals of Mathematics and Artificial Intelligence* **40** (2004) 215–223
9. Priebe, C.E., Marchette, D.J., DeVinney, J.G., Socolinsky, D.A.: Classification using class cover catch digraphs. *Journal of Classification* **20** (2003) 3–23
10. Marchette, D.J.: *Random Graphs for Statistical Pattern Recognition*. John Wiley & Sons (2004)

11. DeVinney, J.G.: The Class Cover Problem and Its Application in Pattern Recognition. Ph.D. Thesis, The Johns Hopkins University (2003)
12. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms 2nd Ed. MIT Press (2001)
13. Cannon, A.H., Ettinger, J.M., Hush, D., Scovel, C.: Machine learning with data dependent hypothesis classes. *Journal of Machine Learning Research* **2** (2002) 335–358
14. Welzl, E.: Smallest enclosing disks (balls and ellipsoids). *New Results and New Trends in Computer Science, LNCS* **555** (1991) 359–370
15. Gärtner, B., Schönher, S.: Fast and robust smallest enclosing balls. *Proc. 7th Annual European Symposium on Algorithms (ESA), LNCS* **1643** (1999) 325–338
16. Fischer, K., Gärtner, B., Kutz, M.: Fast smallest-enclosing-ball computation in high dimensions. *Proc. the 11th Annual European Symposium on Algorithms (ESA)* (2003)
17. Zhou, G.L., Toh, K.C., Sun, J.: Efficient algorithms for the smallest enclosing ball problem. *Computational Optimization and Applications*, (accepted) (2004)
18. Blake, C., Merz, C.: UCI repository of machine learning databases (1998)

SSC: Statistical Subspace Clustering

Laurent Candillier^{1,2}, Isabelle Tellier¹, Fabien Torre¹,
and Olivier Bousquet²

¹ GRAppA - Charles de Gaulle University - Lille 3
candillier@grappa.univ-lille3.fr

² Pertinence - 32 rue des Jeûneurs -75002 Paris
olivier.bousquet@pertinence.com

Abstract. *Subspace clustering* is an extension of traditional *clustering* that seeks to find *clusters* in different subspaces within a dataset. This is a particularly important challenge with high dimensional data where the *curse of dimensionality* occurs. It has also the benefit of providing smaller descriptions of the *clusters* found.

Existing methods only consider numerical databases and do not propose any method for *clusters visualization*. Besides, they require some input parameters difficult to set for the user. The aim of this paper is to propose a new *subspace clustering* algorithm, able to tackle databases that may contain continuous as well as discrete attributes, requiring as few user parameters as possible, and producing an interpretable output.

We present a method based on the use of the well-known *EM algorithm* on a probabilistic model designed under some specific hypotheses, allowing us to present the result as a set of rules, each one defined with as few relevant dimensions as possible. Experiments, conducted on artificial as well as real databases, show that our algorithm gives robust results, in terms of classification and interpretability of the output.

1 Introduction

Clustering is a powerful exploration tool capable of uncovering previously unknown patterns in data [3]. *Subspace clustering* is an extension of traditional *clustering*, based on the observation that different *clusters* (groups of data points) may exist in different subspaces within a dataset. This point is particularly important with high dimensional data where the *curse of dimensionality* can degrade the quality of the results. *Subspace clustering* is also more general than *feature selection* in that each subspace is local to each *cluster*, instead of global to everyone. It also helps to get smaller descriptions of the *clusters* found since *clusters* are defined on fewer dimensions than the original number of dimensions.

Existing methods only consider numerical databases and do not propose any method for *clusters visualization*. Besides, they require some input parameters difficult to set for the user. The aim of this paper is to propose a new *subspace clustering* algorithm, able to tackle databases that may contain continuous as well as discrete attributes, requiring as few user parameters as possible, and

producing an interpretable output. We present a method based on the use of a probabilistic model and the well-known *EM algorithm* [14]. We add in our model the assumption that the *clusters* follow independent distributions on each dimension. This allows us to present the result as a set of rules since dimensions are characterized independently from one another. We then use an original technique to keep as few relevant dimensions as possible to describe each of these rules representing the *clusters*.

The rest of the paper is organized as follows: in section 2, we present existing *subspace clustering* methods and discuss their performances; we then describe our proposed algorithm called **SSC** in section 3; the results of our experiments, conducted on artificial as well as real databases, are then reported in section 4; finally, section 5 concludes the paper and suggests topics for future research.

2 Subspace Clustering

The *subspace clustering* problem has been recently introduced in [2]. Many other methods emerged then, among which two families can be distinguished according to their subspace search method:

1. *bottom-up* subspace search methods [2,6,9,8] that seek to find clusters in subspaces of increasing dimensionality, and produce as output a set of clusters that can overlap,
2. and *top-down* subspace search methods [1,13,15,12,7] that use *k-means like* methods with original techniques of local feature selection, and produce as output a partition of the dataset.

In [10], the authors have studied and compared these methods. They point out that every method requires input parameters difficult to set for the user, and that influence the results (density threshold, mean number of relevant dimensions of the clusters, minimal distance between clusters, etc.). Moreover, although a proposition was made to integrate discrete attributes in *bottom-up* approaches, all experiments were conducted on numerical databases only. Finally, let us note that no proposition was made for producing an interpretable output. This is however crucial because although dimensionality of clusters is reduced in the subspaces specific to them, it can still be too high so that a human user can easily understand it. Yet we will see that in many cases, it is possible to ignore some of these dimensions although keeping the same partition of the data.

The next section presents a new *subspace clustering* algorithm called **SSC**. It is *top-down like* and provides as output a set of clusters represented as rules that may overlap.

3 Algorithm SSC

Let us first denote by N the number of data points of the input database and M the number of dimensions on which they are defined. These dimensions can

be continuous as well as discrete. We suppose values on continuous dimensions are normalized (so that all values belong to the same interval), and denote by $Categories_d$ the set of all possible categories on the discrete dimension d , and $Frequencies_d$ the frequencies of all these categories within the dataset.

3.1 Probabilistic Model

One aim of this paper is to propose a probabilistic model that enables to produce an interpretable output. The basis of our model is the classical mixture of probability distributions $\theta = (\theta_1, \dots, \theta_K)$ where each θ_k is the vector of parameters associated with the k^{th} cluster to be found, denoted by C_k (we set to K the total number of clusters). In order to produce an interpretable output, the use of rules (hyper-rectangles in subspaces of the original description space) is well suited because rules are easily understandable by humans. To integrate this constraint into the probabilistic model, we propose to add the hypothesis that data values follow independent distributions on each dimension. Thus, the new model is less expressive than the classical one that takes into account the possible correlations between dimensions. But it is adapted to the presentation of the partition as a set of rules because each dimension of each cluster is characterized independently from one another. Besides, the algorithm is thus faster than with the classical model because the new model needs less parameters ($O(M)$ instead of $O(M^2)$) and operations on matrices are avoided.

In our model, we suppose data follow gaussian distributions on continuous dimensions and multinomial distributions on discrete dimensions. So the model has the following parameters θ_k for each cluster C_k : π_k denotes its weight, μ_{kd} its mean and σ_{kd} its standard deviation on continuous dimensions d , and $Freq_{kd}$ the frequencies of each category on discrete dimensions d .

3.2 Maximum Likelihood Estimation

Given a set D of N data points \vec{X}_i , *Maximum Likelihood Estimation* is used to estimate the model parameters that best fit the data. To do this, the *EM algorithm* is an effective two-step process that seeks to optimize the *log-likelihood* of the model θ according to the dataset D , $LL(\theta|D) = \sum_i \log P(\vec{X}_i|\theta)$:

1. E-step (*Expectation*): find the class probability of each data point according to the current model parameters.
2. M-step (*Maximization*): update the model parameters according to the new class probabilities.

These two steps iterate until a stopping criterion is reached. Classically, it stops when $LL(\theta|D)$ increases less than a small positive constant δ from one iteration to another.

The E-step consists of computing the membership probability of each data point \vec{X}_i to each cluster C_k with parameters θ_k . In our case, dimensions are assumed to be independent. So the membership probability of a data point to a cluster is the product of membership probabilities on each dimension. Besides,

to avoid that a probability equal to zero on one dimension cancels the global probability, we use a very small positive constant ϵ .

$$P(\vec{X}_i|\theta_k) = \prod_{d=1}^M \max(P(X_{id}|\theta_{kd}), \epsilon)$$

$$P(X_{id}|\theta_{kd}) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma_{kd}}} e^{-\frac{1}{2}\left(\frac{X_{id}-\mu_{kd}}{\sigma_{kd}}\right)^2} & \text{if } d \text{ continuous} \\ \text{Freqs}_{kd}(X_{id}) & \text{if } d \text{ discrete} \end{cases}$$

$$P(\vec{X}_i|\theta) = \sum_{k=1}^K \pi_k \times P(\vec{X}_i|\theta_k) \quad \text{and} \quad P(\theta_k|\vec{X}_i) = \frac{\pi_k \times P(\vec{X}_i|\theta_k)}{P(\vec{X}_i|\theta)}$$

Then the M-step consists of updating the model parameters according to the new class probabilities as follows:

$$\pi_k = \frac{1}{N} \sum_i P(\theta_k|\vec{X}_i)$$

$$\mu_{kd} = \frac{\sum_i X_{id} \times P(\theta_k|\vec{X}_i)}{\sum_i P(\theta_k|\vec{X}_i)} \quad \text{and} \quad \sigma_{kd} = \sqrt{\frac{\sum_i P(\theta_k|\vec{X}_i) \times (X_{id} - \mu_{kd})^2}{\sum_i P(\theta_k|\vec{X}_i)}}$$

$$\text{Freqs}_{kd}(\text{cat}) = \frac{\sum_{\{i|X_{id}=\text{cat}\}} P(\theta_k|\vec{X}_i)}{\sum_i P(\theta_k|\vec{X}_i)} \quad \forall \text{cat} \in \text{Categories}_d$$

It is well known that with the classical stopping criterion, convergence can be slow with *EM*. In order to make our algorithm faster, we propose to add the following *k-means like* stopping criterion: stop whenever the membership of each data point to their most probable cluster does not change. To do this, we introduce a new view on each cluster C_k , corresponding to the set S_k , of size N_k , of data points belonging to it: $S_k = \{\vec{X}_i | \text{Argmax}_{j=1}^K P(\vec{X}_i|\theta_j) = k\}$.

It is also well known that the *EM algorithm* results are very sensitive to the choice of the initial solution. So we run the algorithm many times with random initial solutions and finally keep the model optimizing the *log-likelihood* $LL(\theta|D)$.

At this stage, our algorithm needs one information from the user: the number of clusters to be found. This last parameter of the system can be found automatically with the widely used *BIC* criterion [14]:

$$BIC = -2 \times LL(\theta|D) + m_M \log N$$

with m_M the number of independent parameters of the model. *BIC* criterion must be minimized to optimize the likelihood of the model to the data. So, starting from $K = 2$, the algorithm with fixed K is run and *BIC* is computed. Then K is incremented, and iterations stop when *BIC* increases.

3.3 Output Presentation

To make the results as comprehensible as possible, we now introduce a third view on each cluster corresponding to its description as a rule defined with as few dimensions as possible.

Relevant Dimensions Detection.

In order to select the relevant dimensions of the clusters, we compare on each dimension the likelihood of our model with that of a uniform model. Thus, if the likelihood of the uniform model is greater than the one of our model on one dimension, this dimension is considered to be irrelevant for the cluster. Let us first define the likelihood of a model θ' on a cluster C_k and a dimension d :

$$LL(\theta'|C_k, d) = \sum_{\vec{X}_i \in S_k} \log P(X_{id}|\theta')$$

In the case of a uniform model θ_{U_c} on continuous dimensions, as we suppose the database is normalized, we set $P(X_{id}|\theta_{U_c}) = 1$, and so $LL(\theta_{U_c}|C_k, d) = 0$. Thus, a continuous dimension d is considered to be relevant for a cluster C_k if

$$LL(\theta_{kd}|C_k, d) > 0$$

In the case of discrete dimensions, let θ_{U_d} be the uniform distribution. Then we set $P(X_{id}|\theta_{U_d})=1/|Categories_d|$. So $LL(\theta_{U_d}|C_k, d) = -N_k \times \log |Categories_d|$. For our model on discrete dimensions,

$$LL(\theta_{kd}|C_k, d) = \sum_{\vec{X}_i \in S_k} \log Freks_{kd}(X_{id})$$

As $LL(\theta_{kd}|C_k, d)$ is always greater than $LL(\theta_{U_d}|C_k, d)$ and both are negative, we need to introduce a constant $0 < \alpha < 1$ and set that d is relevant for the cluster if

$$LL(\theta_{kd}|C_k, d) > \alpha \times LL(\theta_{U_d}|C_k, d)$$

Dimension Pruning.

Although we have already selected a subset of dimensions relevant for each cluster, it is still possible to prune some and simplify the clusters representation while keeping the same partition of the data.

See figure 1 as an example. In that case, the cluster on the right is dense on both dimensions X and Y . So its true description subspace is $X \times Y$. However, we do not need to consider Y to distinguish it from the other clusters: define it by high values on X is sufficient. The same reasoning holds for the cluster on the top.

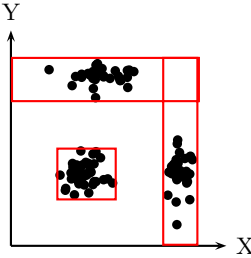


Fig. 1. Example of minimal description

To do this dimension pruning, we first create the rule R_k associated with the current cluster C_k . We now only consider the set of dimensions considered as relevant according to the previous selection. On continuous dimensions, we associate with the rule the smallest interval containing all the coordinates of the data points belonging to S_k . For discrete dimensions, we chose to associate with the rule the most probable category.

We then associate a weight W_{kd} with each dimension d of the rule R_k . For continuous dimensions, it is the ratio between local and global standard deviation according to μ_{kd} . And for discrete dimensions, it is the relative frequency of the most probable category.

$$W_{kd} = \begin{cases} 1 - \frac{\sigma_{kd}^2}{\sigma_d^2}, \text{ with } \sigma_d^2 = \frac{\sum_i (X_{id} - \mu_{kd})^2}{N} & \text{if } d \text{ continuous} \\ \frac{Freqs_{kd}(cat) - Frequencies_d(cat)}{1 - Frequencies_d(cat)} & \text{if } d \text{ discrete} \\ \text{with } cat = Argmax_{\{c \in Categories_d\}} Freqs_{kd}(c) & \end{cases}$$

We then compute the support of the rule (the set of data points comprised in the rule). This step is necessary since it is possible that some data points belong to the rule but not to the cluster. And finally, for all relevant dimensions presented in ascending order of their weights, delete the dimension from the rule if the deletion does not modify its support.

4 Experiments

Experiments were conducted on artificial as well as real databases. The first ones are used to observe the robustness of our algorithm faced with different types of databases. In order to compare our method with existing ones, we conducted these experiments on numerical-only databases. Then real databases are used to show the effectiveness of the method on real-life data (that may contain discrete attributes).

4.1 Artificial Databases

Artificial databases are generated according to the following parameters: N the number of data points in the database, M the number of (continuous) dimensions on which they are defined, K the number of clusters, MC the mean dimensionality of the subspaces on which the clusters are defined, SD_m and SD_M the minimum and maximum standard deviation of the coordinates of the data points belonging to a same cluster, from its centroid and on its specific dimensions.

K random data points are chosen on the M -dimensional description space and used as seeds of the K clusters (C_1, \dots, C_K) to be generated. Let us denote them by $(\vec{O}_1, \dots, \vec{O}_K)$. With each cluster is associated a subset of the N data points and a subset (of size close to MC) of the M dimensions that will define its specific subspace. Then the coordinates of the data points belonging to a

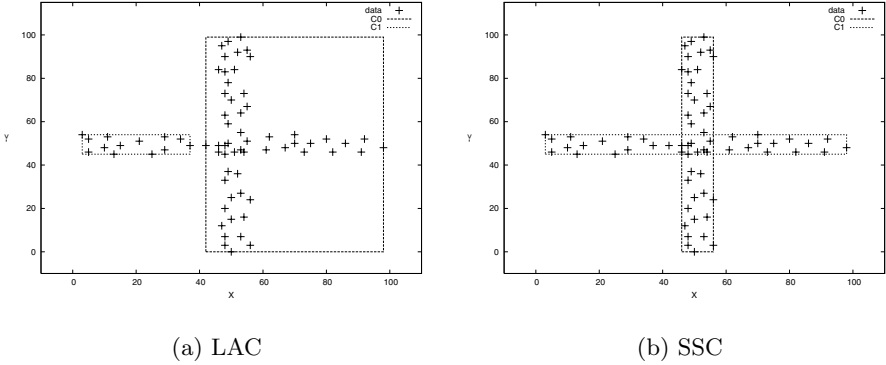


Fig. 2. LAC versus SSC

cluster C_k are generated according to a normal distribution with mean O_{kd} and standard deviation $sd_{kd} \in [SD_m..SD_M]$ on its specific dimensions d . They are generated uniformly between 0 and 100 on the other dimensions.

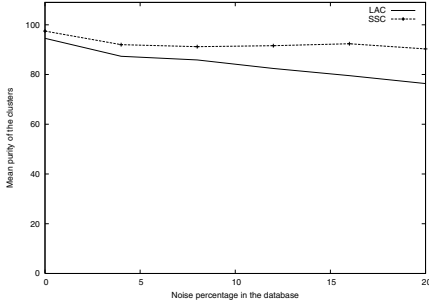
Our method is *top-down like*. Among the most recent ones, LAC [7] is an effective method that, as ours, only needs one user parameter: the number of clusters to be found (if we do not use BIC). So we propose to compare our method with LAC and provide to both algorithms the number of clusters to be found. LAC is based on *k-means* and associates with each centroid a vector of weights on each dimension. At each step and for each cluster, these weights on each dimension are updated according to the dispersion of the data points of the cluster on the dimension (the greater the dispersion, the less the weight).

Figure 2 shows the result of LAC and **SSC** on an artificial database. On this example, we can observe a classical limitation of *k-means like* methods over *EM like* methods: the first ones do not accept that data points belong to multiple clusters whereas the second ones give to each data point a membership probability to each cluster. Thus, contrary to *EM like* methods, *k-means like* methods are not able to capture concepts like the one appearing in figure 2 (one cluster is defined on one dimension and takes random values on another, and conversely for the other one) because of the intersection between clusters.

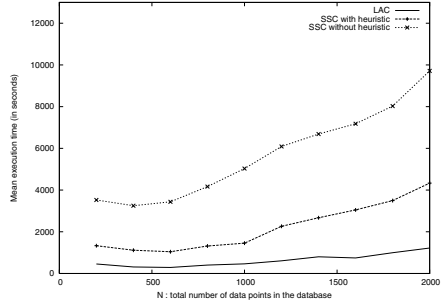
Experiments conducted on artificial databases with different generation parameters pointed out the robustness of our method. In particular, we observe that it is resistant to noise (see figure 3(a)). Accuracy of the partition is measured by the average purity of the clusters (the purity of a cluster is the maximum percentage of data points belonging to the same initial concept). With 20% of noise in the database, the average purity of the clusters is 90 for **SSC** while only 76 for LAC.

Our method is also robust to missing values. When summing over all data values on one dimension, the only thing to do is to ignore the missing values.

Concerning the execution time of our algorithm, experiments pointed out that the acceleration heuristic we proposed in section 3.2 is effective: for results



(a) Resistance to noise varying between 0 and 20%.

(b) Execution time according to $200 < N < 2000$.**Fig. 3.** Artificial tests with $N = 600$, $M = 30$, $K = 5$, $MC = 3$, $SD_m = 2$, $SD_M = 5$

of the same quality, the computing times of **SSC** with the heuristic are nearer to that of LAC (k -means like methods are well known for their efficiency) than to that of **SSC** without the heuristic (see figure 3(b)).

Let us finally note that the results of our method are still robust even if data were generated by uniform distributions inside given intervals on the specific dimensions of the clusters, instead of normal distributions.

4.2 Real Databases

Experiments were also conducted on real databases. Among them, the *Automobile* database coming from UCI repository [4] contains the description of 205 cars defined by a mix of 16 continuous and 10 discrete attributes. On this database, the three clusters found by **SSC** are characterized with a mean of only four dimensions. It thus points out that our method is effective in reducing the dimensionality, and thus giving an interpretable description of the clusters found.

Besides, this reduced description also allows us to compute with few cost a weight associated with each couple of relevant dimensions corresponding to the visualization power of this couple (remind W_{ki} is the weight, for the cluster C_k , of the dimension i):

$$V_{ij} = \sum_{k=1}^K \max(W_{ki}, W_{kj})$$

The graphical visualizations corresponding to the two more visual couples of dimensions in the case of the *Automobile* database are provided figure 4. It thus visually shows that the price of cars increases a lot when their length exceeds 170 (figure 4(a)), that the cars with *rear-wheel drive* (*rwd*) have an average higher *curb-weight* than cars with *front-wheel drive* and *4-wheel drive* (figure 4(b)), and that the majority of the most expensive cars are *rear-wheel drive* (correspondance between both figures concerning cluster C_2).

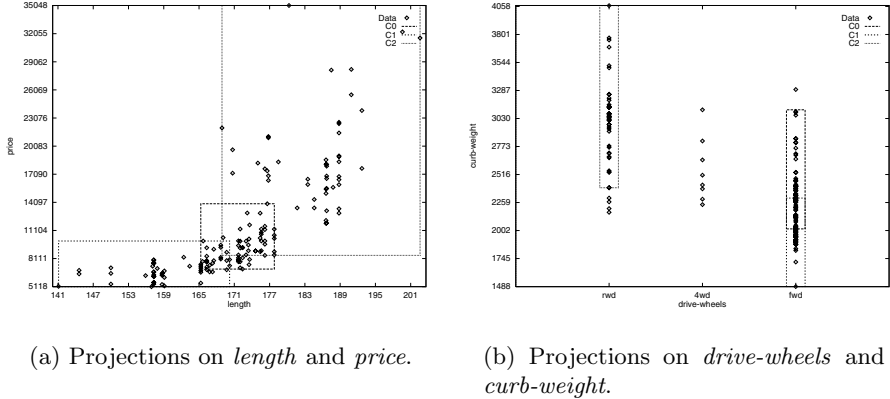


Fig. 4. Results of SSC on the *Automobile* database for $K = 3$

5 Conclusion

We have presented in this paper a new *subspace clustering* method based on the use of a probabilistic model with the specific assumption that data were following independent distributions on each dimension. This idea has already been studied in [11]. But the method described by the authors differs from ours on some points. First, instead of using a mixture of gaussians on continuous dimensions, they use a mixture of uniform density M-dimensional hyper-rectangles supplemented with gaussian “tails”, depending on a parameter σ that decreases during execution. Thus, their method is not adapted for incremental learning, whereas **SSC** can update its model when new data points arise. Moreover, we effectively integrated the problem of handling discrete dimensions whereas it was just mentioned as potential improvements in [11]. We have also proposed an original technique of dimension selection allowing us to provide as output an interpretable and visual representation of the clusters found.

Besides, we have proposed an original heuristic to speed up our algorithm. To continue our investigation in that direction, it seems interesting to take into account the work of [5] that is about the acceleration of the *EM algorithm* in the general case. Another way should be to consider only relevant dimensions during the iteration process.

Finally, we think it can be interesting to adapt our method for supervised or semi-supervised learning. And it should also be interesting to study the effectiveness of our method in a feature selection task.

References

1. Charu C. Aggarwal, Joel L. Wolf, Philip S. Yu, Cecilia Procopiuc, and Jong Soo Park. Fast algorithms for projected clustering. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 61–72, 1999.

2. Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 94–105, Seattle, Washington, 1998.
3. Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, California, 2002.
4. C.L. Blake and C.J. Merz. UCI repository of machine learning databases [<http://www.ics.uci.edu/~mllearn/MLRepository.html>], 1998.
5. P. Bradley, U. Fayyad, and C. Reina. Scaling EM (Expectation-Maximization) clustering to large databases. Technical report, Microsoft Research, Aug. 1998.
6. Chun Hung Cheng, Ada Wai-Chee Fu, and Yi Zhang. Entropy-based subspace clustering for mining numerical data. In *Knowledge Discovery and Data Mining*, pages 84–93, 1999.
7. Carlotta Domeniconi, Dimitris Papadopoulos, Dimitrios Gunopulos, and Sheng Ma. Subspace clustering of high dimensional data. In *SIAM Int. Conf. on Data Mining*, 2004.
8. Karin Kailing, Hans-Peter Kriegel, and Peer Kröger. Density-connected subspace clustering for high-dimensional data. In *SIAM Int. Conf. on Data Mining*, pages 246–257, 2004.
9. Harsha Nagesh, Sanjay Goil, and Alok Choudhary. Mafia: Efficient and scalable subspace clustering for very large data sets. Technical report, Northwestern University, 1999.
10. Lance Parsons, Ehtesham Haque, and Huan Liu. Evaluating subspace clustering algorithms. In *Workshop on Clustering High Dimensional Data and its Applications*, *SIAM Int. Conf. on Data Mining*, pages 48–56, 2004.
11. Dan Pelleg and Andrew Moore. Mixtures of rectangles: Interpretable soft clustering. In Carla Brodley and Andrea Danyluk, editors, *18th Int. Conf. on Machine Learning*, pages 401–408. Morgan Kaufmann, San Francisco, California, 2001.
12. Ioannis A. Sarafis, Phil W. Trinder, and Ali M. S. Zalzala. Towards effective subspace clustering with an evolutionary algorithm. In *IEEE Congress on Evolutionary Computation*, Canberra, Australia, Dec. 2003.
13. Kyoung-Gu Woo and Jeong-Hoon Lee. *FINDIT: a fast and intelligent subspace clustering algorithm using dimension voting*. PhD thesis, Korea Advanced Institute of Science and Technology, Department of Electrical Engineering and Computer Science, 2002.
14. L. Ye and M.E. Spetsakis. Clustering on unobserved data using mixture of gaussians. Technical report, York University, Toronto, Canada, Oct. 2003.
15. Kevin Y. Yip, David W. Cheung, and Michael K. Ng. A highly-usable projected clustering algorithm for gene expression profiles. In *3rd ACM SIGKDD Workshop on Data Mining in Bioinformatics*, pages 41–48, 2003.

Understanding Patterns with Different Subspace Classification

Gero Szepannek, Karsten Luebke, and Claus Weihs*

Department of Statistics, University of Dortmund
szepannek@statistik.uni-dortmund.de

Abstract. By identifying *characteristic regions* in which classes are dense and also relevant for discrimination a new, intuitive classification method is set up. This method enables a visualized result so the user is provided with an insight into the data with respect to discrimination for an easy interpretation. Additionally, it outperforms Decision trees in a lot of situations and is robust against outliers and missing values.

1 Introduction

Classification or supervised learning often involves two goals: the first is allocation or prediction, i.e. assigning class labels to new observations. The second goal, which can be even more important, is descriptive and involves the discovery of the underlying differences between the classes. The new Different Subspace Classification (DiSCo) method is a method to simultaneously visualize and classify multi-class problems in high dimensional spaces and is therefore designed to attain both predictive and descriptive goals.

Decision trees and Naive Bayes classifiers are two of the most often used data mining techniques. In case of Decision trees this may be due to the fact that the result of a tree can often be interpreted in terms of the subject matter (see e.g. Hastie et al. 2001, p. 267). Furthermore, Decision trees perform variable selection: variables which are not relevant for classification are not used to build the tree. A shortcoming of trees is that in the final tree only parts of the marginal distribution of the variables are used, conditional on the split. Another major problem caused by the hierarchical structure of a tree is the inherent instability to small changes in the data resulting in high variance (Hastie et al. 2001, p.274). To overcome this, Random Forests (Breiman 2001) and Bagging (Breiman 1996) can be applied but then the easy interpretation is lost. The Naive Bayes method is somewhat different. There, all class-conditional univariate marginal densities are estimated independently. Especially in high dimensional feature spaces Naive Bayes often performs well (see e.g. Hastie et al. 2001, p. 185). Unfortunately the result of Naive Bayes is not so easy to interpret and it can not be used to select variables. Also it is not robust against outliers.

* This work has been supported by the Collaborative Research Center 475 of the German Research Foundation (DFG).

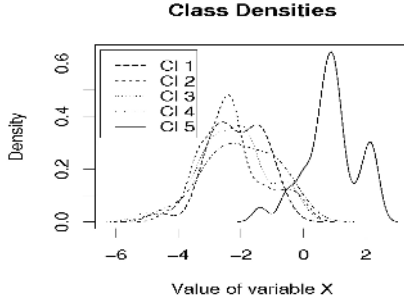


Fig. 1. Density estimation of five classes

The higher the dimension of the data the more challenging is the understanding of the data. So if there are many observed variables, methods of variable selection are often used to reduce the dimension of the data. Such methods identify and retain those of the variables that separate the classes best – like a Decision tree does. Afterwards, a classification method is (re-)applied to the resulting subspace of variables. A problem may be that in general the variables do not contain relevant separating-information for all classes. So, a variable can contain information for separating class i from the rest but no information for the discrimination of class $j \neq i$. This may be illustrated by Figure 1. Density estimation of five classes is shown and it can be seen that by this variable, an object of class 5 may be probably well separated from the others. But a value of e.g. -2 will not tell us much about its real class (which may be probably one of the classes 1 to 4). The new DiSCo method can be considered as a mixture of both Decision trees and Naive Bayes: calculate all class-conditional univariate marginal densities by an appropriate kind of histogram estimate comparable to Naive Bayes and find out the so called *dense* regions, where many objects of a class fall in. Check whether these regions are *relevant* to distinguish this class from others and take away all dispensable information like a Decision tree. Both ideas together are used in the new method to tackle classification problems. Moreover, it can be seen that it is robust against outliers, missing values and can be used with metric and categorical data. In DiSCo variable selection is intrinsic to the classification method. The resulting subsets of variables which are used for discrimination of the classes can differ between the classes. Another focus of the new classification method lies on the visualization of the class-characteristics. The proposed method does not make any assumptions about the underlying distribution of the data. The only, not very strong assumption is that objects of the same class are similar in some of their observed variables.

In the following section the concept of *characteristic regions* is defined and a classification rule is developed. Section 3 explains the visualization of the results. Section 4 briefly summarizes the choice of parameters for the implementation of the method while section 5 contains results of a comparative study of the three mentioned methods on simulated data.

2 Notation and Method

The idea of the new method is to search for characteristic regions, i.e. sets of values in some variables that indicate the class-membership. To build up these characteristic regions two steps are needed. The first step is to search for intervals of the realizations of the random variables that contain a large probability mass of the classes. The resulting "regions" are called dense regions. The second step, which is independent of the first, identifies regions that discriminate at least one class from the others because of a relatively high density. These regions are called relevant regions. Regions that are both dense and relevant are then called characteristic regions.

2.1 Characteristic Regions

The concept of the characteristic regions is given as follows:

Definition 1.

- For metric variables X^d (where d is the variable index):
 S^d being the set of all possible realizations of an object x_n in variable X^d , for each d let $\{R_m^d : 0 \leq m \leq M^d + 1\}$ be a contiguous segmentation of an interval covering S^d following

1. $\bigcup_{m=0}^{M^d+1} R_m^d \supseteq S^d$
 (All possible realizations of X^d are covered by the union of all its regions.)
2. $\forall x_1, x_2 \in R_m^d$ and $\alpha \in [0, 1] : \alpha x_1 + (1 - \alpha)x_2 \in R_m^d$
 (The regions of every variable are contiguous.)
3. $\forall x_1 \in R_{m_1}^d, x_2 \in R_{m_2}^d, m_1 < m_2 : x_1 < x_2$
 (In every variable the regions are disjoint and also ordered.)

R_m^d are called **regions** of variable X^d . $M^d(+2)$ denotes the number of regions variable X^d . A possible choice is proposed in section 4.1.

By restriction 2 all the objects that fall into one region can be considered to be similar.

- For categorical variables X^d :
 If a variable is categorical the **regions** are implicitly given by all its possible values. Sometimes it may be reasonable for the user to merge some of the values to one region if there are too many levels or because of the subject matter.

Definition 2. Let x_n^d be the value taken by object n in variable X^d and let k_n be the corresponding, known index of its class. Then

$$n_m^d(k) := \sum_{n=1}^N I_{[R_m^d]}(x_n^d) I_{[k]}(k_n) \quad (1)$$

with $I_{[\cdot]}$ as the indicator function is called the **corresponding frequency** of class k in Region m of variable d .

As the $n_m^d(k)$ should represent the density of the data it is assumed for simplicity of comparisons that for any fixed d and all $1 \leq m \leq M^d$:

$$\sup_{x \in R_m^d} - \inf_{x \in R_m^d} \equiv const.,$$

so the regions of a variable have equal width. $m = \{0, M^d + 1\}$ are necessary to form "outer regions" (see section 4.1). By this the corresponding frequencies are proportional to heights of histogram bars of the classes if the bandwidths are given by the regions.

Let **dense regions** be those regions which contain most of the classes' probability masses. Let $S_{DR} > 0$ be a threshold to construct class wise dense regions. Then, dense regions are regions $R_{m_0}^d(k)$ with

$$n_{m_0}^d(k) \geq S_{DR} \frac{\sum_{m=0}^{M^d+1} n_m^d(k)}{M^d} \tag{2}$$

This proceeding corresponds to comparing the observed corresponding frequency to the mean over all regions.

Relevant regions should be the regions where the density of one class k is high compared to those of the other classes and so a new observed object lying in this region strongly indicates its membership to class k . Let $S_{RR} > 0$ be a threshold to construct class-wise relevant regions. Then, relevant regions are regions $R_{m_0}^d(k_0)$ with:

$$\frac{n_{m_0}^d(k_0)}{N_{k_0}} \geq S_{RR} \frac{\sum_{k=1}^K \frac{n_{m_0}^d(k)}{N_k}}{K} \tag{3}$$

with K being the number of different classes. To be able to compare the regions' densities of different classes by corresponding frequencies they have to be weighted by their observed absolute frequencies N_k . Finally, **characteristic regions** are regions that are both dense and relevant.

Missing values in one or more variables can simply be omitted when building the (variable-wise) regions without loss of information for the other variables.

2.2 Classification Rule

Let $w_m^d(k) \geq 0$ be a **class wise weight of a region** of class k connected to region R_m^d .

The characteristic regions are used to build up the classification rule by summing the weights over all variables. Then the assignment of the class is obtained by

$$\hat{k}(x_{new}) = \arg \max_k \sum_{d=1}^D \sum_{m=0}^{M^d+1} I_{[R_m^d]}(x_{new}) w_m^d(k) \tag{4}$$

where the weights of the characteristic regions are defined by

$$w_m^d(k_0) := \begin{cases} 0 & \text{if (2) or (3) do not hold} \\ \frac{n_m^d(k_0) \frac{p(k_0)N}{N_{k_0}}}{\sum_{k=1}^K n_m^d(k) \frac{p(k)N}{N_k}} & \text{if } R_m^d \text{ is characteristic for class } k_0 \end{cases} \quad (5)$$

where $\frac{p(k)N}{N_k}$ adjusts the *corresponding frequencies* if the observed class frequencies differ from known a priori class probabilities $p(k)$. The weights are motivated by the marginal probability of $k_{new} = k$ given $x_{new}^d \in R_m^d$, if R_m^d is "characteristic" for class k .

3 Visualization

The weights $w_m^d(k)$ described above mimic marginal conditional probability of the different classes. As only characteristic regions will be shown in our visualization only robust information relevant for classification is given. So plotting these class wise weights of the regions (see equation 5) provides a visualization of the class characteristics and an interpretation may be simplified.

As example we illustrate the method in Figure 2 on the well known Iris data set. The values of the variables are shown on the x-axes while the different colours of the bars symbolize the different true classes (black = "Setosa", light

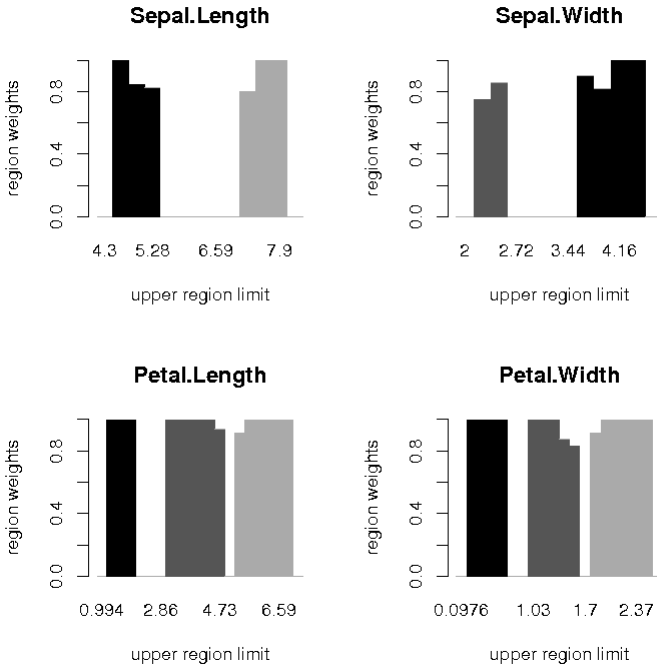


Fig. 2. Example: Visualization of a result for Iris data

grey = "Virginica" and dark grey = "Versicolor"). The heights are the weights of the characteristic regions. It can be seen that the variable "Sepal length" only serves to indicate membership of one of the classes "Virginica" or "Setosa" but not for "Versicolor", while the variable "Sepal width" just serves to characterize a plant of class "Setosa" or "Versicolor". The "Petal" variables seem to separate all three classes with the lowest values for class "Setosa". The upper extreme values indicate the class "Virginica". As the plots of these two variables are of the same structure one can suppose a correlation between these variables.

4 Implementation of the Method

4.1 Building the Regions for Metric Variables

As mentioned earlier the *corresponding frequencies* are proportional to heights of histogram bars for simplicity, so we can refer to the theory of nonparametric density estimation to build the regions. In histogram density estimation a problem consists in smoothing but not over-smoothing the empirical distribution of the data. Thus the bin-width of a histogram should be chosen neither too small nor too large. Freedman et al. 1981 suggest a choice of

$$bw = \frac{2}{\sqrt[3]{N}} IQR \quad (6)$$

as bin-width where IQR is the interquartile range. Under weak assumptions this histogram is L^2 -convergent for density estimation (Freedman et al. 1981). As the distribution may be different in the classes this bin-width calculation must be done for every class and every variable separately, returning $bw(k, d)$.

The number of class-wise bins is then $M^d(k) = r\left(\frac{x_{(N_k)}^d - x_{(1_k)}^d}{bw(k, d)}\right)$ with $x_{(N_k)}^d$ and $x_{(1_k)}^d$ being the class-wise maximum or minimum, respectively, $r(\cdot)$ being the rounding operator. With $IV^d := [x_{(1)}^d, x_{(N)}^d]$ and $IV_k^d := [x_{(1_k)}^d, x_{(N_k)}^d]$ let:

$$M^d := r\left(\left\{\sum_k \left(M^d(k) \int_{IV_k^d} \left\{\sum_k I_{[IV_k^d]}(s)\right\}^{-1} ds\right)\right\} * \frac{\int_{IV^d} 1 dt}{\int_{\cup_k IV_k^d} 1 dt}\right) \quad (7)$$

In the simplest case, if the densities of the classes do not overlap and also there is no interspace between them, (7) reduces to $M^d = \sum_k M^d(k)$. If there is space between the classes this space must be filled also with bins. Therefore, in (7) with $\int_{IV^d} 1 dt$, the width of the whole interval IV^d , is related to $\int_{\cup_k IV_k^d} 1 dt$, the width of those parts of the whole interval which are covered by values of the different classes. If there is some free space between the classes this is smaller than the whole interval and the number of bins is linearly projected. In cases when densities of the classes do overlap this must also be corrected. By the calculation of $\left\{\sum_k I_{[IV_k^d]}(s)\right\}^{-1}$ it is assured that those parts of IV_k^d which are covered by more classes than just class k are not repeatedly counted in the calculation of M^d . So the class wise number of bins is linearly projected or

averaged, respectively, for intervals covered by $0, 1, 2, \dots$ classes. In other words M^d is a linear projection of the k class-wise number of bins $M^d(k)$ of the parts covered by the different classes to the whole range of the data averaged for the classes' overlap. The regions of variable d are IV^d divided into M^d equal parts. R_0^d and $R_{M^d+1}^d$ cover the upper and lower rest.

By construction DiSCo is robust to outliers as outliers are not dense by definition. The only problem that may occur is that an unnecessarily high value for the number of bins is calculated. Therefore class wise outliers can be removed in the calculation of the number of bins without any loss of information. Outliers can be for example all observations that differ more than a fixed value (for example $\frac{3}{2}$ times the standard deviation) from the variables' median.

4.2 Optimizing the Thresholds

There remains the question how to choose the thresholds in equation 2 and equation 3. So far no theoretical background is known for an optimal choice of both S_{DR} (dense regions) and S_{RR} (relevant regions). The optimal parameters are found by a 2-dimensional grid-search algorithm. As the criterion for optimization the cross validated error rate (on training data) is used. Concerning the parameters one can suppose that a rather small threshold S_{DR} eliminates outliers but keeps a large probability mass in the remaining regions. S_{RR} rather large keeps only regions in the model that strongly indicate one class.

5 Benchmark Study

In the previous sections, we focussed on classification methods that work on the variables separately, namely Classification trees, Naive Bayes and the newly developed DiSCo method. We will now compare these methods in a quite general simulation study to investigate the advantages of each method.

We will start with the simple case of normally distributed data in the following subsection. Then, we turn to situations where the assumption of normality is violated. Subsection 5.2 simulates situations with multimodality in the data and in subsection 5.3 the effect of outliers is investigated.

5.1 Normally Distributed Data

We simulated data consisting of three classes and three variables. Each class is separated from the other classes by a different mean in one variable – while the other two classes have the same mean in that variable (compare Figure 1 where the mean of class 5 is separated from classes 1-4). All variables are normally distributed with variance 1. The location difference is chosen to be twice the α -quantile of the standard normal distribution, guaranteeing a controlled probability of overlap and therefore misclassification of the classes. So the Bayes risk for the separated class is α . For the other two classes with same means the expected error rate equals 0.5 so that a random choice will be as good. All

Table 1. Test error rates on normally distributed data at varying probabilities of class-overlap

Class overlap	Naive Bayes	CART	DiSCo
0.010	0.001	0.021	0.016
0.050	0.017	0.065	0.043
0.100	0.063	0.109	0.099
0.400	0.525	0.572	0.578

simulations rely on 300 objects in both training and test data set each class having the same prior probability. The results are averaged over all 30 repeated simulations.

Table 1 shows the effect of the classes' overlap in normally distributed data on the misclassification rate for the different methods. Of course, since the assumption of normality holds Naive Bayes turns out to have lowest error rates. For large location differences (i.e. small overlapping probabilities) all methods show very small error rates, as expected. In such situations, DiSCo is preferable to Classification trees since the error rates of the trees are up to 50% higher compared to those of DiSCo. We also tested deviation from normality by different skewness levels, but this had almost no influence the performance of the methods compared to each other so these results are omitted here.

5.2 Effect of Multimodality

Another violation of normality may be caused by multimodality of the data. This seems to be an important case for practical applications since classes may consist of several different "subclasses", leading to multimodal distributions. We constructed data as in section 5.1 but with each class possessing a bimodal distribution. The distributions are designed as follows: an object is with probability $p = 0.5$ from one of two normal distributions $N(0, 1)$ or $N(2*\alpha, 1)$. In each of the three variables two of the classes are identically distributed following the bimodal distribution specified before. The third class differs in location to both others in a manner that the two underlying distributions are shifted to be $N(-\alpha, 1)$ or $N(\alpha, 1)$. α is varied to investigate different levels of overlap of the classes. It determines the overlap of two neighbouring modes and is varied as in section 5.1. The results (Table 2) show unacceptably large error rates when wrongly assuming (unimodal) normally distributed data as for the Naive Bayes method. The DiSCo error rates dominate those of the Classification tree. With increasing overlap the performance of Naive Bayes is approximating those of the the other methods. The visualization of the results of the three different methods is shown in Figure 3. The Decision tree visualizes the whole decision rule and is therefore maybe the most comprehensive way to display the entire decision. Nevertheless, since there are many conditional splits in the tree, the specific characteristics of the three classes are hardly identifiable. The results of Naive Bayes and the DiSCo method can be visualized for each variable separately. For Naive Bayes,

Table 2. Test error rates on bimodal data at varying overlap percentages between two neighbouring densities

% overlap	Naive Bayes	CART	DiSCo
0.001	0.379	0.010	0.004
0.050	0.386	0.125	0.094
0.100	0.406	0.232	0.192
0.200	0.432	0.416	0.401

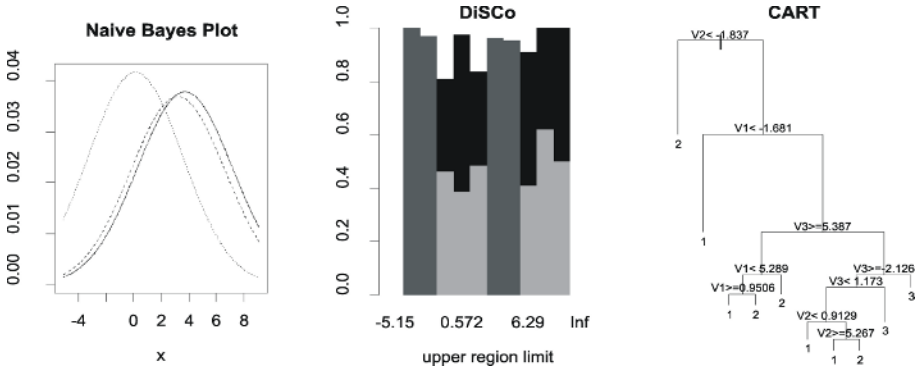


Fig. 3. Visualization of the results of the different methods on bimodally distributed data, only one variable is shown for Naive Bayes and DiSCo

one can plot the density estimation of the different classes. This here gives a completely wrong impression of the structure of the classes, while DiSCo (as introduced in section 3) nicely displays the classes' characteristics: all modes are identified and it can be seen whether their locations are characteristic for one single or more than one class.

5.3 Effect of Outliers

The last study investigates the behavior of the three methods if the data are contaminated with outliers. The data sets are generated as in section 5.1 except that with chance of 5% an object is an outlier following a distribution $N(0, \sigma^2)$ with variance much larger than those of the classes' distributions. The classification errors are examined for different sizes of σ .

Classification trees and DiSCo behave relatively robust to outlier contamination while Naive Bayes becomes much worse with increasing variance of the outliers. We further observe that the misclassification rates of Classification trees are systematically worse than those of DiSCo.

Table 3. Test error rates on outlier contaminated data at varying variance of the outlier distribution

SD of outliers	Naive Bayes	CART	DiSCo
3	0.032	0.070	0.054
10	0.052	0.079	0.057
20	0.099	0.077	0.059
50	0.331	0.078	0.067

6 Summary

Motivated by the fact that different regions in the variable may discriminate some but not all classes a new classification method is set up. By identifying *characteristic regions* that indicate whether regions of values are dense and also relevant for discrimination this method implicitly includes a feature selection. Moreover, it is robust to outliers and missing values in the observed data. Also the descriptive aspect of data analysis is addressed by an informative visualization of the DiSCo result.

A benchmark study is performed where the new method is compared to Classification trees and the Naive Bayes classifier since both methods also work on the marginal data. Different situations are examined. Comparing the missclassification rates, the Naive Bayes classifier performs better than both other classifiers if the assumption of normality holds while DiSCo has smaller error rates than the Classification tree. If the data are generated from multimodal distributions or contaminated with outliers, Naive Bayes' error rates become unacceptably high. The other two methods are able to handle such data while the misclassification rates of the Classification trees are slightly dominated by those of DiSCo.

References

- Breiman, L.: Bagging predictors. *Machine Learning* **26** (1996) 123–140
- Breiman, L.: Random forests. *Machine Learning* **45** (2001) 5–32
- Breiman, L., Friedman, J., Olshen, R. and Stone, C.: *Classification and regression trees*. Wadsworth Publishing Co Inc. 1984.
- Freedman, D. and Diaconis, P.: On the histogram as a density estimator: L_2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* **67** (1981) 453–476
- Hastie, T., Tibshirani, R. and Friedman, J.: *The elements of statistical learning*. Springer. 2001.

Using Clustering to Learn Distance Functions for Supervised Similarity Assessment

Christoph F. Eick, Alain Rouhana, Abraham Bagherjeiran,
and Ricardo Vilalta

Department of Computer Science, University of Houston,
Houston, TX 77204-3010, USA
{ceick, rouhana, vilalta, abagherj}@cs.uh.edu

Abstract. Assessing the similarity between objects is a prerequisite for many data mining techniques. This paper introduces a novel approach to learn distance functions that maximizes the clustering of objects belonging to the same class. Objects belonging to a data set are clustered with respect to a given distance function and the local class density information of each cluster is then used by a weight adjustment heuristic to modify the distance function so that the class density is increased in the attribute space. This process of interleaving clustering with distance function modification is repeated until a “good” distance function has been found. We implemented our approach using the k-means clustering algorithm. We evaluated our approach using 7 UCI data sets for a traditional 1-nearest-neighbor (1-NN) classifier and a compressed 1-NN classifier, called NCC, that uses the learnt distance function and cluster centroids instead of all the points of a training set. The experimental results show that attribute weighting leads to statistically significant improvements in prediction accuracy over a traditional 1-NN classifier for 2 of the 7 data sets tested, whereas using NCC significantly improves the accuracy of the 1-NN classifier for 4 of the 7 data sets.

1 Introduction

Many tasks, such as case-based reasoning, cluster analysis and nearest-neighbor classification, depend on assessing the similarity between objects. Defining object similarity measures is a difficult and tedious task, especially in high-dimensional data sets.

Only a few papers center on learning distance function from training examples. Stein and Niggemann [10] use a neural network approach to learn weights of distance functions based on training examples. Another approach, used by [7] and [9], relies on an interactive system architecture in which users are asked to rate a given similarity prediction, and then uses reinforcement learning to enhance the distance function based on the user feedback.

Other approaches rely on an underlying class structure to evaluate distance functions. Han, Karypis and Kumar [4] employ a randomized hill-climbing approach to learn weights of distance functions for classification tasks. In their

approach k-nearest-neighbor queries are used to evaluate distance functions; the k-neighborhood of each object is analyzed to determine to which extend the class labels agree with the class label of each object. Zhihua Zhang [13] advocates the use of kernel functions and multi-dimensional scaling to learn Euclidean metrics. Finally, Hastie et. al. [5] propose algorithms that learn adaptive rectangular neighborhoods (rather than distance functions) to enhance nearest-neighbor classifiers.

There has also been some work that has some similarity to our work under the heading of semi-supervised clustering. The idea of *semi-supervised clustering* is to enhance a clustering algorithm by using side information that usually consists of a "small set" of classified examples. Xian's approach [12] transforms the classified training examples into constraints: points that are known to belong to different classes need to have a distance larger than a given bound. He then derives a modified distance function that minimizes the distance between points in the data set that are known to belong to the same class with respect to these constraints using classical numerical methods ([1] advocates a somewhat similar approach). Klein [6] proposes a shortest path algorithm to modify a Euclidean distance function based on prior knowledge.

This paper introduces an approach that learns distance functions that maximize class density. It is different from the approaches that were discussed above in that it uses clustering and not k-nearest-neighbor queries to evaluate a distance function; moreover, it uses reinforcement learning and not randomized hill climbing or other numerical optimization techniques to find "good" weights of distance functions.

The paper is organized as follows. Section 2 introduces a general framework for similarity assessment. Section 3 introduces a novel approach that learns weights of distance functions using clusters for both distance function evaluation and distance function enhancement. Section 4 describes our approach in more depth. Section 5 discusses results of experiments that analyze the benefits of using our approach for nearest-neighbor classifiers. Finally, Section 6 concludes the paper.

2 Similarity Assessment Framework Employed

In the following a framework for similarity assessment is proposed. It assumes that objects are described by sets of attributes and that the similarity of different attributes is measured independently. The dissimilarity between two objects is measured as a weighted sum of the dissimilarity with respect to their attributes. To be able to do that, a weight and a distance measure has to be provided for each attribute. More formally, define:

$O = \{o_1, \dots, o_n\}$	Set of objects whose similarity has to be assessed
$o_{i,j}$	Value of attribute att_j for object $o_i \in O$
Θ_i	Distance function of the i -th attribute
w_i	Weight for the i -th attribute

Based on the definitions in the above table, the distance Θ between two objects o_1 and o_2 is computed as follows:

$$\Theta(o_1, o_2) = \frac{\sum_{i=1}^m w_i \Theta_i(o_{1,i}, o_{2,i})}{\sum_{i=1}^m w_i}$$

3 Interleaving Clustering and Distance Function Learning

In this section, we will give an overview of our distance function learning approach. Then, in the next section, our approach is described in more detail. The key idea of our approach is to use clustering as a tool to evaluate and enhance distance functions with respect to an underlying class structure. We assume that a set of classified examples is given. Starting from an initial object distance function d_{init} , our goal is to obtain a “better” distance function d_{good} that maximizes class density in the attribute space.

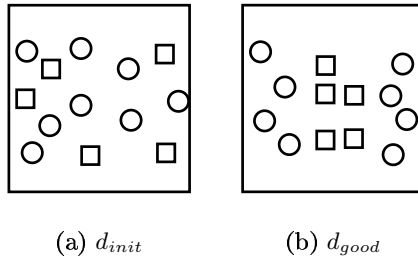


Fig. 1. Visualization of the Objectives of the Distance Function Learning Process

Fig. 1 illustrates what we are trying to accomplish; it depicts the distances of 13 examples, 5 of which belong to a class that is identified by a square and 8 belong to a different class that is identified by a circle. When using the initial distance function d_{init} we cannot observe too much clustering with respect to the two classes; starting from this distance function we like to obtain a better distance function d_{good} so that the points belonging to the same class are clustered together. In Fig. 1 we can identify 3 clusters with respect to d_{good} , 2 containing circles and one containing squares. Why is it beneficial to find such a distance function d_{good} ? Most importantly, using the learnt distance function in conjunction with a k-nearest-neighbor classifier allows us to obtain a classifier with high predictive accuracy. For example, if we use a 3-nearest-neighbor classifier with d_{good} it will have 100% accuracy with respect to leave-one-out cross-validation, whereas several examples are misclassified if d_{init} is used. The second advantage is that looking at d_{good} itself will tell us which features are important for the particular classification problem.

There are two key problems for finding “good” object distance functions:

1. We need an evaluation function that is capable of distinguishing between good distance functions, such as d_{good} , and not so good distance functions, such as d_{init} .
2. We need a search algorithm that is capable of finding good distance functions.

Our approach to address the first problem is to cluster the object set O with respect to the distance function to be evaluated. Then, we associate an error with the result of clustering process that is measured by the percentage of minority examples that occur in the clusters obtained.

Our approach to the second problem is to adjust the weights associated with the i -th attribute relying on a simple reinforcement learning algorithm that employs the following weight adjustment heuristic. Let us assume a cluster contains 6 objects whose distances with respect to att_1 and att_2 are depicted in Fig. 2:



Fig. 2. Idea Underlying the Employed Weight Adjustment Approach

If we look at the distribution of the examples with respect to att_1 we see that the average distance between the majority class examples (circles in this case) is significantly smaller than the average distance considering all six examples that belong to the cluster; therefore, it is desirable to increase the weight w_1 of att_1 , because we want to drive the square examples "into another cluster" to enhance class purity; for the second attribute att_2 the average distance between circles is larger than the average distance of the six examples belonging to the clusters; therefore, we would decrease the weight w_2 of att_2 in this case. The goal of these weight changes is that the distances between the majority class examples are decreased, whereas distances involving non-majority examples are increased. We will continue this weight adjustment process until we processed all attributes for each cluster; then we would cluster the examples again with the modified distance function (as depicted in Fig. 3), for a fixed number of iterations.

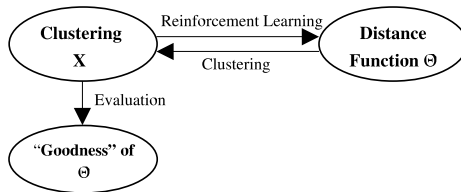


Fig. 3. Co-evolving Clusters and Distance Functions

4 Using Clusters for Weight Learning and Distance Function Evaluation

Before we can introduce our weight adjustment algorithm, it is necessary to introduce the notations in Table 1 that are later used when describing our algorithms.

As discussed in [4], searching for good weights of distance functions can be quite expensive. Therefore in lieu of conducting a “blind” search for good weights, we like to use local knowledge, such as density information within particular clusters, to update weights more intelligently. In particular, our proposed approach uses the average distances between the majority class members¹ of a cluster and the average distance between all members belonging to a cluster for the purpose of weight adjustment. More formally, let:

w_i be the current weight of the i -th attribute

σ_i be the average normalized distances for the examples that belong to the cluster with respect to Θ_i

μ_i be the average normalized distances for the examples of the cluster that belong to the majority class with respect to Θ_i

Then the weights are adjusted with respect to a particular cluster using formula W:

$$w'_i = w_i + \alpha w_i (\sigma_i - \mu_i) \quad (1)$$

with $1 \geq \alpha > 0$ being the learning rate.

Table 1. Notations used in the description of our algorithms

O	Set of objects (belonging to a data set)
c	Number of different classes in O
$n = O $	Number of objects in the data set
D_i	Distance matrix with respect to the i -th attribute
D	Object distance matrix for O
$X = \{c_1, \dots, c_k\}$	Clustering of O with each cluster C_i being a subset of O
$k = X $	Number of clusters used
$\pi(\Theta, O)$	Clustering algorithm that computes a set of clusters X
$\Psi(\Theta, O) = q(\pi(\Theta, O))$	Evaluation function for Θ using a clustering algorithm π
$q(X)$	Evaluation function that measures the impurity of a clustering

In summary, after a clustering² has been obtained with respect to a distance function the weights of the distance function are adjusted using formula 1 iterating over the obtained clusters and the given set of attributes. It should also

¹ If there is more than one most frequent class for a cluster, one of those classes is randomly selected to be “the” majority class of the cluster.

² Clusters are assumed to be disjoint.

be noted that no weight adjustment is performed for clusters that are pure or for clusters that only contain single examples belonging to different classes.

Example 1. Assume we have a cluster that contains 6 objects numbered 1 through 6 with objects 1, 2, 3 belonging to the majority class. Furthermore, we assume there are 3 attributes with three associated weights w_1, w_2, w_3 which are assumed to be equal initially ($w_1 = w_2 = w_3 = \frac{1}{3}$) and distance matrices $D_1, D_2,$ and D_3 with respect to the 3 attributes are given below; e.g. object 2 has a distance of 2 to object 4 with respect to Θ_1 , and a distance of 3 to object 1 with respect to Θ_3 :

$$\begin{matrix}
 D_1 & D_2 & D_3 & D \\
 \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 3 \\ & 0 & 1 & 2 & 3 & 1 \\ & & 0 & 1 & 2 & 2 \\ & & & 0 & 1 & 3 \\ & & & & 0 & 1 \\ & & & & & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 1 & 1 & 5 & 1 \\ & 0 & 1 & 1 & 5 & 1 \\ & & 0 & 1 & 5 & 5 \\ & & & 0 & 5 & 1 \\ & & & & 0 & 5 \\ & & & & & 0 \end{bmatrix} & \begin{bmatrix} 0 & 3 & 3 & 2 & 2 & 3 \\ & 0 & 3 & 2 & 2 & 3 \\ & & 0 & 2 & 2 & 2 \\ & & & 0 & 1 & 3 \\ & & & & 0 & 1 \\ & & & & & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1.67 & 2 & 2 & 3.67 & 2.33 \\ & 0 & 1.67 & 1.67 & 3.33 & 1.67 \\ & & 0 & 1.33 & 3 & 3 \\ & & & 0 & 2.33 & 2.33 \\ & & & & 0 & 2.33 \\ & & & & & 0 \end{bmatrix}
 \end{matrix}$$

The object distance matrix D is next computed using:

$$D = \frac{w_1 D_1 + w_2 D_2 + w_3 D_3}{w_1 + w_2 + w_3}$$

First, the average cluster and average inter-majority object distances for each attribute have to be computed; we obtain: $\sigma_1 = 2, \mu_1 = 1.3; \sigma_2 = 2.6, \mu_2 = 1; \sigma_3 = 2.2, \mu_3 = 3$. The average distance and the average majority examples distance within the cluster with respect to Θ are: $\sigma = 2.29, \mu = 1.78$. Assuming $\alpha = 0.2$, we obtain the new weights: $\{1.14*0.33333, 1.32*0.33333, 0.84*0.33333\}$. After the weights have been adjusted for the cluster, the following new object distance matrix D is obtained:

$$D = \begin{bmatrix} 0 & 1.51 & 1.86 & 1.95 & 3.89 & 2.20 \\ & 0 & 1.51 & 1.60 & 3.55 & 1.51 \\ & & 0 & 1.26 & 3.20 & 3.20 \\ & & & 0 & 2.60 & 2.20 \\ & & & & 0 & 2.60 \\ & & & & & 0 \end{bmatrix}$$

After the weights have been adjusted for the cluster, the average inter-object distances have changed to: $\sigma = 2.31, \mu = 1.63$. As we can see, the examples belonging to the majority class have moved closer to each other (the average majority class example distance dropped by 0.15 from 1.78), whereas the average distances of all examples belonging to the cluster increased very slightly, which implies that the distances involving non-majority examples (involving objects 4, 5 and 6 in this case) have increased, as intended.

The weight adjustment formula we introduced earlier gives each cluster the same degree of importance when modifying the weights. If we had two clusters,

Table 2. Data Sets Used in the Experimental Evaluation

Dataset	n	Classes	Attributes
DIABETES	768	2	8
VEHICLE	846	4	18
HEART-STATLOG	270	2	13
GLASS	214	6	9
HEART-C	303	5	13
HEART-H	294	5	13
IONOSPHERE	351	2	34

one with 10 majority examples and 5 minority examples, and the other with 20 majority and 10 minority examples, with both clusters having identical average distances and average majority class distances with respect to the attributes, the weights of attributes would have identical increases (decreases) for the two clusters. This somehow violates common sense; more efforts should be allocated to remove 10 minority examples from a cluster of size 30, than to removing 5 members of a cluster that only contains 15 objects. Therefore, we add a factor λ to our weight adjustment heuristic that makes weight adjustment somewhat proportional to the number of minority objects in a cluster. Our weight adjustment formula therefore becomes:

$$w'_i = w_i + \alpha \lambda w_i (\sigma_i - \mu_i)$$

with λ being defined as the number of minority examples in the cluster over the average number of minority examples per clusters.

For example, if we had 3 clusters that contain examples belonging to 3 different classes with the following class distributions (9, 3, 0), (9, 4, 4), (7, 0, 4); the average number of minority examples per cluster in this case is $(3+8+4)/3=5$; therefore, λ would be $3/5=0.6$ when adjusting the weights of the first cluster, $8/5$ when adjusting the weights of the second cluster, and $4/5$ when adjusting the weights in the third cluster.

As explained earlier, our approach searches for good weights using a weight adjustment heuristic that was explained in the previous section; however, how do we know which of the found distance functions is the best? In our approach a distance function is evaluated based on the impurity of the clustering X obtained with respect to the distance function to be evaluated. As explained earlier, impurity is measured by the percentage of minority examples that occur in the different clusters of solution X . A minority example is an example that belongs to a class different from the most frequent class in its cluster.

5 Experimental Evaluation

5.1 Data Sets Used and Preprocessing

We tested the distance function learning approach for a benchmark consisting of the following 7 datasets: DIABETES, VEHICLE, HEART-STATLOG, GLASS,

HEART-C, HEART-H, and IONOSPHERE that were taken from the University of Irvine's Machine Learning Benchmark [2]. Table 2 gives a short summary for each data set. All seven data sets only contain numerical attributes. The numerical attributes in each data set were normalized by using a linear interpolation function that assigns 1 to the maximum value and 0 to the minimum value for that attribute in the data set.

5.2 Algorithms Evaluated in the Experiments

In the experiments conducted, we compared the performance of two different 1-nearest-neighbor classifiers that use the learnt weights with a traditional 1-nearest-neighbor classifier that considers all attributes to be of equal importance. Moreover, we also compare the results with a decision tree classifier. Details about the algorithm evaluated will be given in this section.

Our distance function learning approach does not only learn a distance function Θ , but also obtains a centroid and a majority class for each cluster. These (centroid, majority class) pairs can be used to construct a 1-nearest-neighbor classifier that we call *nearest centroid classifier* (NCC) in the following. NCC is based on the idea that a cluster's centroid is used as the representative for a cluster. NCC classifies new examples by assigning the majority class of the closest centroid to it. NCC uses the learnt distance function Θ to determine the closest centroid. A nearest centroid classifier can be viewed as a "compressed" 1-nearest-neighbor classifier that operates on a set of $k < n$ cluster representatives, rather than using all training examples.

In particular, in the experiments the following four algorithms were tested for the 7 data sets that have been described in the previous section:

- 1-NN.** 1-nearest-neighbor classifier that uses all examples of the training set and does not use any attribute weighting.
- LW1NN.** 1-nearest-neighbor classifier with attribute weighting (same as 1-NN but weights are learnt using the methods we described in Sections 3 and 4)
- NCC.** 1-nearest-neighbor classifier that uses k (centroid, majority class) pairs, instead of all objects in the data set; it also uses attribute weighting.
- C4.5.** uses the C4.5 decision tree learning algorithm that is run with its default parameter settings.

5.3 Experimental Results

Experiments were conducted using the WEKA toolkit [11]. The accuracy of the four algorithms was determined by running 10-fold cross validation 10 times. Table 2 shows the accuracy results averaged over the ten runs of cross validation for each data set/classification algorithm pair.

The weight learning algorithm was run for 200 iterations and best weight combination found with respect to q was reported. We used $1/j$ (where j is the number of attributes) as the initial weights; that is, attributes are assumed to have the "same" importance initially. Moreover, after each iteration weights were

normalized so that the sum of all weights always adds up to 1. The learning rate α was linearly decreased from 0.6 at iteration 1 to 0.3 at iteration 200.

A supervised clustering algorithm [3] was used to determine the k-values for the DIABETES, and VEHICLE data sets, and for the other data sets k-values were set to 5 times the number of classes in the data set. The decision tree and 1-NN classifiers used in the experiments are the standard classifiers that accompany the WEKA toolkit. The remaining algorithms use two modified WEKA algorithms: the k-means clustering and 1-NN algorithms. The modifications to each permit the use of attribute weights when computing object similarity.

We chose the 1-NN classifier as the reference algorithm for the experiments and indicated statistically significant improvements³ of other algorithms over 1-NN in bold face in Table 2. The table also indicates the number of objects n in each data set, as well as the parameter k that was used when running k-means. If we compare the 1-nearest-neighbor classifier with our attribute weighting approach (LW1NN), we see that the weight learning approach demonstrates significant improvements of more than 3.5% in accuracy for the GLASS, and IONOSPHERE data sets (also outperforming C4.5 for those data sets), but does not show any statistically significant improvements for the other five data sets.

Table 3. Accuracy for the 4 Classification Algorithms

Dataset	n	k	1-NN	LW1NN	NCC	C4.5
DIABETES	768	35	70.62	68.89	73.07	74.49
VEHICLE	846	64	69.59	69.86	65.94	72.28
HEART-STATLOG	270	10	76.15	77.52	81.07	78.15
GLASS	214	30	69.95	73.5	66.41	67.71
HEART-C	303	25	76.06	76.39	78.77	76.94
HEART-H	294	25	78.33	77.55	81.54	80.22
IONOSPHERE	351	10	87.1	91.73	86.73	89.74

Using NCC, on the other hand, demonstrates significant improvements in accuracy for the DIABETES, HEART-STATLOG, HEART-C, and HEART-H data sets, which is quite surprising considering the small number of representatives used by the compressed classifier. For example, for the HEART-C data set the 303*0.9 objects⁴ in a training set were replaced by 25 (centroid, majority class)-pairs and the distance function that considers every attribute of equal importance was replaced with the learnt distance function. Moreover, the accuracies of the three HEART data sets are at least 1.5% higher than those reported for C4.5. Again, not surprisingly, for the other data sets reducing the number

³ Statistical significance was determined by a paired t-test on the accuracy for each of the 10 runs of 10-fold cross validation.

⁴ 10-fold cross validation only uses 90% of the examples of a data set for training.

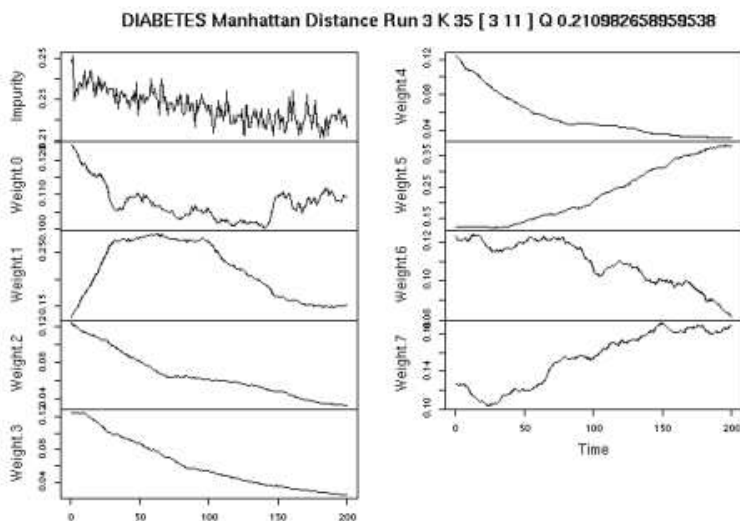


Fig. 4. Display of a run of the weight Learning Algorithm for the VEHICLE Data Set

of objects used by a nearest-neighbor classifier, results in a reduced accuracy, sometimes significantly less.

Each run of the weight learning process for each fold was captured through a set of graphs. Fig. 4 depicts one of those run summaries for the DIABETES data set. It shows how cluster impurity and weights changed during that particular run. The DIABETES data set has eight attributes and therefore eight weights Weight0,...,Weight7 that are initially set to 0.125. The initial impurity was about 25% at the beginning of the run and the minimum impurity of 21% was reached approximately at iteration 180; the other graphs depict how the 8 weights of the 8 attributes changed iteration by iteration. For example, Weight3 dropped from its initial value of 0.125 to approximately 0.03 at the end of the run, whereas Weight5 increased from 0.125 to approximately 0.38 near the end of the run. As mentioned earlier, weights are normalized prior to clustering; therefore, the sum of the 8 weights always adds up to 1.

6 Conclusion

The paper presented a novel approach to learn distance functions with respect to an underlying class structure that uses clusters for both distance function evaluation and distance function enhancement. We also proposed a novel weight adjustment heuristic that adapts weights of distance functions based on class density information in clusters.

The proposed weight learning approach was evaluated using a benchmark consisting of 7 data sets. Our empirical results suggest that our attribute weight-

ing approach enhanced the prediction accuracy of a 1-nearest-neighbor (1-NN) classifier significantly for 2 of the 7 data sets. However, our approach does not only learn a distance function, but also provides a centroid and a majority class for each cluster. These (centroid, majority class) pairs can be used to construct a nearest centroid classifier (NCC) that is a "compressed" nearest-neighbor classifier. Surprisingly, although it uses only a small percentage of the examples belonging to a data set using NCC lead to significant improvements in accuracy for 4 of the 7 data sets.

We claim that our approach facilitates the tailoring of distance functions supporting different perspectives with respect to the data set to be analyzed. For example, one user might be interested in analyzing products with respect to the time spend for their delivery whereas another analyst is interested in analyzing products with respect to the profit they produced. Using our approach we would generate a distance function for each analysis perspective that, we believe, is particularly important for data warehousing applications, such as those that rely on OLAP-technology.

References

1. Bar-Hillel, A., Hertz, T., Shental, N. & Weinshall, D. (2003). Learning Distance Functions Using Equivalence Relations, in Proc. ICML'03, Washington D.C.
2. Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases [<http://www.ics.uci.edu/~mllearn/MLRepository.html>]Irvine, CA: University of California, Department of Information and Computer Science.
3. Eick, C., Zeidat, N. (2005). *Using Supervised Clustering to Enhance Classifiers*, in Proc. 15th International Symposium on Methodologies for Intelligent Systems (ISMIS), Saratoga Springs, New York.
4. Han, E.H., Karypis, G. & Kumar, V. (1999). Text Categorization Using Weight Adjusted nearest-neighbor Classification, Lecture Notes in Computer Science.
5. Hastie, T. & Tibshirani, R. (1996). Discriminant Adaptive Nearest-Neighbor Classification, in IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, pp. 607-616.
6. Klein, D., Kamvar, S.-D. & Manning, C. (2002). From instance-level Constraints to Space-level Constraints: Making the Most of Prior Knowledge in Data Clustering, in Proc. ICML'02, Sydney, Australia.
7. Kira K. & Rendell L. (1992). A practical Approach to Feature Selection, in Proc. 9th Int. Conf on Machine Learning.
8. MacQueen, J (1967). Some methods for classification and analysis of multi-variate observations, in Proc. 5th Berkeley Symposium Math., Stat., Prob., 1:281-297.
9. Salzberg, S. (1991). A nearest Hyperrectangle Learning Method, Machine Learning.
10. Stein, B. & Niggemann, O. (2001). Generation of Similarity Measures from Different Sources, in Proc. Fourteenth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems Springer Verlag.
11. Witten, I. & Eibe, F. (2000). *Data Mining: Practical machine learning tools with Java implementations*, by Ian H. Witten and Eibe Frank, Morgan Kaufmann, San Francisco.

12. Xing, E.P., Ng, A., Jordan, M. & Russell S. (2003). Distance Metric Learning with Applications to Clustering with Side Information. *Advances in Neural Information Processing* 15, MIT Press.
13. Zhang, Z. (2003). Learning Metrics via Discriminant Kernels and Multi-Dimensional Scaling: Toward Expected Euclidean Representation, in *Proc. ICML'03*, Washington D.C.

Linear Manifold Clustering

Robert Haralick and Rave Harpaz

Pattern Recognition Laboratory,
The Graduate Center, City University of New York,
365 Fifth Avenue New York, NY 10016, USA
haralick@ptah.gc.cuny.edu
rbharpaz@sci.brooklyn.cuny.edu

Abstract. In this paper we describe a new cluster model which is based on the concept of linear manifolds. The method identifies subsets of the data which are embedded in arbitrary oriented lower dimensional linear manifolds. Minimal subsets of points are repeatedly sampled to construct trial linear manifolds of various dimensions. Histograms of the distances of the points to each trial manifold are computed. The sampling corresponding to the histogram having the best separation between a mode near zero and the rest is selected and the data points are partitioned on the basis of the best separation. The repeated sampling then continues recursively on each block of the partitioned data. A broad evaluation of some hundred experiments over real and synthetic data sets demonstrates the general superiority of this algorithm over any of the competing algorithms in terms of stability, accuracy, and computation time.

1 Introduction

The problem of clustering can be loosely defined as the partitioning of a set of points in a multidimensional space into groups (*clusters*) such that the points in each group are similar to one another. Finding these clusters is important because their points correspond to observations of different classes of objects that may have been previously unknown. A second kind of latent information that may be of interest, are correlations in a data set. A correlation is a linear dependency between two or more attributes of the data set. Knowing about the existence of a relationship between attributes may enable us to learn hidden causalities. For example, the influence of the age of a patient and the dose rate of medication on the length of his disease.

Due to recent technology advances in data collection many applications of clustering are now characterized by high dimensional data, some of whose dimensions are non-information carrying. Thus, clusters or correlations may be visible only in linear combinations of subsets of the dimensions. Conventional clustering algorithms such as K-means [10], and DBSCAN [8] are "full-dimensional" in the sense that they give equal relevance to all dimensions, and therefore are likely to fail when applied to such high-dimensional data. Subspace clustering is an extension to traditional clustering in that it attempts to find clusters embedded

in different subspaces of the same data set. A subspace cluster consists of a subset of points and a corresponding subset of attributes (or linear combinations of attributes), such that these points form a dense region in a subspace defined by the set of corresponding attributes. Most subspace clustering methods such as CLIQUE [3], MAFIA [11], and PROCLUS [1] are restricted to finding clusters in subspaces spanned by some subset of the original measurement features. However, examination of real data often shows that points tend to get aligned along arbitrarily oriented subspaces. ORCLUS [2] the most relevant algorithm to our problem is an extension to PROCLUS which allows clusters to exist in arbitrarily oriented subspaces. Dasgupta [5] presents two important results related to *random projections* which have implications to clustering in high dimensional spaces. These results show that it is possible to project high dimensional data into substantially lower dimensions while still retaining the approximate level of separation between clusters. In a recent paper Haralick et al. [6] use random projections in the context of projection pursuit to search for interesting one dimensional projections that reveal inter-cluster separations. Their algorithm, called HPCluster, uses an hierarchical approach that repeatedly bi-partitions the data set using interesting one-dimensional projections.

In this paper we describe a new cluster model that is based on the concept of linear manifolds. It takes into account both linear dependencies among features and distances between points. In section 2 we formalize our model of a cluster. Based on this model, we present in section 3 the algorithm-LMCLUS. In section 4 we present a broad evaluation of LMCLUS applied on synthetic and real data sets, and in section 5 we conclude the paper giving hints on future work.

2 The Cluster Model

The goal is to find clusters with an intrinsic dimensionality that is much smaller than the dimensionality of the data set, and that exhibit correlation among some subset of attributes or linear combinations of attributes. The cluster model which we propose has the following properties: the points in each cluster are embedded in a lower dimensional linear manifold ¹. The intrinsic dimensionality of the cluster is the dimensionality of the linear manifold. The manifold is arbitrarily oriented. The points in the cluster induce a correlation among two or more attributes (or linear combinations of attributes) of the data set. In the orthogonal complement space to the manifold the points form a compact densely populated region.

Definition 1 (Linear Manifold). *L is a **linear manifold** of vector space V if and only if for some subspace S of V and translation $t \in V$, $L = \{x \in V | \text{for some } s \in S, x = t + s\}$. The dimension of L is the dimension of S .*

¹ A linear manifold is a translated subspace. A subspace is a subset of points closed under linear combination.

Definition 2 (Linear Manifold Cluster Model). Let D be a set of d -dimensional points, $C \subseteq D$ a subset of points that belong to a cluster, x some point in C , b_1, \dots, b_l an orthonormal set of vectors that span \mathbb{R}^d , (b_i, \dots, b_j) a matrix whose columns are the vectors b_i, \dots, b_j , and μ some point in \mathbb{R}^d . Then each $x \in C$ is modeled by,

$$x = \mu + (b_1, \dots, b_l)\lambda + (b_{l+1}, \dots, b_d)\psi, \tag{1}$$

where μ is the cluster mean, λ is a zero mean random $l \times 1$ vector whose entries are i.i.d. $U(-R/2, +R/2)$, ψ is a zero mean random vector with small variance independent of λ , and R is the range of the data.

The idea is that each point in a cluster lies close to an l -dimensional linear manifold, which is defined by $\mu + span\{b_1, \dots, b_l\}$. It is easy to see that μ is the cluster mean since

$$E[x] = E[\mu + (b_1, \dots, b_l)\lambda + (b_{l+1}, \dots, b_d)\psi] =$$

$$\mu + (b_1, \dots, b_l)E[\lambda] + (b_{l+1}, \dots, b_d)E[\psi] = \mu + (b_1, \dots, b_l)\mathbf{0} + (b_{l+1}, \dots, b_d)\mathbf{0} = \mu$$

Classical clustering algorithms such as K-means take $l = 0$ and therefore omit the possibility that a cluster has a non-zero dimensional linear manifold associated with it. In the manifold we assume the points are uniformly distributed in each direction according to $U(-R/2, +R/2)$. It is in this manifold that the cluster is embedded, and therefore the intrinsic dimensionality of the cluster will be l . The third component models a small disturbance, or error factor associated with each point in the manifold. The idea is that each point may be perturbed in directions that are orthogonal to the manifold, i.e., the vectors b_{l+1}, \dots, b_d . We model this behavior by requiring that ψ be a $(d-l) \times 1$ random vector, normally distributed according to $N(\mathbf{0}, \Sigma)$, where the largest eigenvalue of Σ is much smaller than R . Since the variance along each of these directions is much smaller than the range R of the embedding, the points are likely to form a compact and densely populated region, which can be used to cluster the data.

Figure 1 is an example of data set modeled by eq. (1). The data set contains three non-overlapping clusters, where C_1, C_2 which are almost planar are embedded in 2D manifolds. Their points are uniformly distributed in the manifold

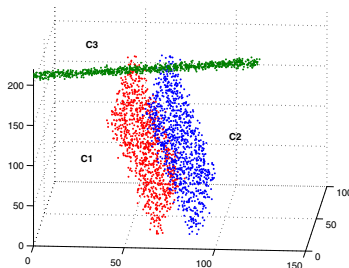


Fig. 1. A data set demonstrating the concept of linear manifold clusters

and they include a random error element in the orthogonal complement space to the manifold. Similarly, C_3 an elongated line like cluster, is embedded in a 1D linear manifold.

3 The Algorithm

LMCLUS can be viewed as an hierarchical-divisive procedure, which marries the ideas of random projection via sampling and histogram thresholding, in order to detect clusters embedded in lower dimensional linear manifolds. It expects three inputs: L , an estimate of the highest dimension of the manifolds in which clusters may be embedded. \hat{K} , an estimate of the largest number of clusters expected to be found, which is used to compute the number of trial manifolds of a given dimensionality that will be examined in order to reveal the best possible partitioning of the data set. Γ , a sensitivity threshold which is used to determine whether or not a partitioning should take place. We note that unlike related methods \hat{K} does not impose a restriction on the number of clusters the algorithm actually finds. The output of LMCLUS is a set of labeled clusters together with the intrinsic dimensionality of each cluster. Knowing the dimensionality associated with each cluster can then be used with methods such as PCA to model the data in each cluster. The algorithm operates by detecting one cluster at a time and successively reapplying itself on the remaining set of points. It iterates over a range of manifold dimensionalities, in an a priori fashion, starting from the lowest-1, and terminating with the highest- L . For each dimensionality the algorithm invokes a procedure which we call *FindSeparation* in an attempt to reveal separations among subsets of the data. Its underlying idea is to successively randomly sample subsets of points that can define a linear manifold of a given dimension. Of the linear manifolds constructed, the one closest to a substantial number of data points is selected. The proximity of the input data points to the manifold is captured by a distance histogram. If the manifold indeed has some subset of points near it, then the distance histogram will reveal a mixture of two distributions. One of the distributions has a mode near zero and is the distribution of distances of points that potentially belong to a cluster, and the other is the distribution of the remaining points in the data set. The problem of separating the cluster points from the rest is then cast into a histogram thresholding problem, which is solved using Kittler and Illingworth minimum error thresholding technique [9]. *FindSeparation* returns four values γ - which is a measure of the “goodness” of the separation, τ - a proximity threshold that is computed from the histogram and is used to split the data, B - the basis of the manifold which exposed the separation, and x_0 -the origin of the manifold. When γ exceeds the value of the sensitivity threshold Γ , indicating that a worthy separation has been found, then the data set is split according to τ . This split corresponds to the partitioning of all the points which are located close enough to a manifold, i.e. all points that potentially belong to a given cluster, and those that belong to other clusters. In addition the dimensionality of the manifold which revealed the separation, corresponding to the intrinsic dimensionality of

the cluster is recorded. An attempt to further partition the cluster which may consist of sub-clusters is executed by reapplying *FindSeparation* until the cluster can not be further separated. At this point the algorithm will attempt to partition the cluster in higher dimensions, a process which will continue until the dimension limit L is reached. When L is reached we have a subset of points that cannot be partitioned any more, and declare that a cluster is found. We note that if outliers exist then the last partition will contain this set of points. By definition outliers do not belong to any cluster and therefore will remain the last group of points to be associated to any other group. Moreover, since they are unlikely to form any clusters the algorithm will not be able to partition them, and therefore will all be grouped together.

Sampling Linear Manifolds. To construct an l -dimensional linear manifold by sampling points from the data we need to sample $l + 1$ points. Let x_0, \dots, x_l denote these points. We choose one of the points x_0 as the origin. Then the l vectors spanning the manifold are obtained by $x'_i = x_i - x_0$ where $i = 1 \dots l$. Assuming each of these sampled points came from the same cluster, then according to eq. (1)

$$x'_i = (\mu_0 + B\lambda_i + B_c\psi_i) - (\mu_0 + B\lambda_0 + B_c\psi_0) = B(\lambda_i - \lambda_0) + B_c(\psi_i - \psi_0)$$

where $B = (b_1, \dots, b_l)$ and $B_c = (b_{l+1}, \dots, b_d)$. If the cluster points did not have an error component, that is, they all lie at distance zero from the manifold, then sampling any $l + 1$ which are linearly independent, and belong to the same cluster would enable us to reconstruct B . Therefore in order to get a good approximation of B we would like each of the sampled points to come from the same cluster, and to be as close as possible to the linear manifold spanned by the column vectors of B . In other words we would like each of the $l + 1, \dots, d$ components of each x'_i to be close to zero, and this occurs when $\psi_i - \psi_0 \approx \mathbf{0}$. A good indication as to why this is likely to occur when the sampled points come from the same cluster, is given by the fact that $E[\psi_i - \psi_0] = \mathbf{0}$. Therefore the problem of sampling a linear manifold that will enable us to separate a cluster from the rest of the data basically reduces to the problem of sampling $l + 1$ points that all come from the same cluster.

Assuming the data set contains K clusters all having approximately the same number of points. Then the probability that a sample of $l + 1$ points all come from the same cluster is approximately $(\frac{1}{K})^l$. The probability that out of n samples of $l + 1$ points, none come from the same cluster, is approximately $(1 - (1/K)^l)^n$ and $1 - (1 - (1/K)^l)^n$ will be the probability that at least for one of the samples all of its $l + 1$ points come from the same cluster. Therefore the sample size n required such that this probability is greater than some value $1 - \epsilon$ is given by

$$n \geq \frac{\log \epsilon}{\log(1 - (1/K)^l)} \quad (2)$$

Thus, by computing n given ϵ , and $K = \hat{K}$ we can approximate a lower bound on the number samples required or trial manifolds that will be examined. Note that by varying \hat{K} we can tradeoff accuracy with efficiency.

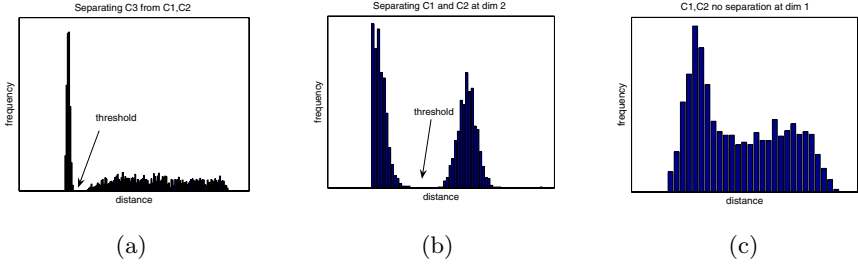


Fig. 2. Histograms used to separate the clusters from Fig. 1. (a) C_3 is separated from C_2 and C_3 by sampling 1D linear manifolds. (b) C_1 is separated from C_2 by sampling 2D linear manifolds. (c) a histogram for which no separation can be found

For each sample of points (x'_1, \dots, x'_l) we construct an orthonormal basis B of a linear manifold, measure distances to it, and then using Kittler and Illingworth’s method we compute a threshold τ . Of all possible thresholds corresponding to different linear manifolds we prefer the one which induces the best separation. That is to say, the one which induces the largest *discriminability* given by $\frac{(\mu_1(\tau) - \mu_2(\tau))^2}{\sigma_1(\tau)^2 + \sigma_2(\tau)^2}$, and the one which causes the deepest broadest minimum in the Kittler and Illingworth criterion function J [9]. This can be measured by the difference/depth of the criterion function evaluated at τ and the value evaluated at the closest local maxima τ' , i.e., *depth* = $J(\tau') - J(\tau)$. Thus, our composite measure of the “goodness” of a separation is given by

$$\gamma = \text{discriminability} \times \text{depth} \quad (3)$$

A set of typical histograms generated during the clustering process are depicted in Fig. 2, corresponding to a subset of the histograms used to cluster the data set given in Fig. 1.

4 Empirical Evaluation

LMCLUS as well as three other related methods: DBSCAN a representative of the full-dimensional clustering methods, ORCLUS a representative of subspace clustering methods, and HPCluster a random projection based clustering method, were implemented in C++. The aim of the experiment was to evaluate LMCLUS’s performance in comparison to the other methods with respect to accuracy, efficiency, scalability and its stability as a stochastic algorithm.

4.1 Synthetic Data Generation

In order to generate clusters embedded in different arbitrary oriented linear manifolds of various dimensions, and following the model given by eq. (1) we used a method similar to one described in the ORCLUS paper. The underlying

idea is to first generate the clusters in an axis parallel manner and then randomly translate and rotate each cluster to achieve the effect of an arbitrary orientation in the space. A candidate data set that we would like to produce is one in which the clusters are relatively close in some subspace with minimal overlap, and yet sparse enough that canonical clustering algorithms would not be able to detect. We also used the *cluster sparsity coefficient* proposed in the ORCLUS paper to measure the relative hardness of clustering a given data set, and selected only data sets which yielded a measure within a specific range.

4.2 Accuracy

To measure the accuracy of LMCLUS we have generated several dozen synthetic data sets of various sizes, space/manifold dimensionalities, and number of clusters. Table 1 summarizes the properties of fifteen representative data sets, along with the performance of each of the algorithms when applied to these data sets. The ones marked with a star ('*') denote *star* data sets due to their star like geometry, which are likely to present difficulties to many clustering algorithms. Accuracy was measured by means of a *Confusion Matrix*. In addition the amount of time (in hours, minutes, and seconds) required by each algorithm to cluster a data set was recorded. These results clearly demonstrate LMCLUS's superiority over the other clustering algorithms. LMCLUS was the only one able to discover (over 85% accuracy) all the clusters. DBSCAN's poor performance emphasizes the ineffectiveness of distance metrics that utilize the full space. Note that only LMCLUS and ORCLUS were able to handle the *star* clusters. However requiring the number of clusters and the dimensionality of the subspaces in which the clusters are embedded makes ORCLUS impractical for real data sets. The fact that HPCluster was not able to cluster the *star* data sets also comes at no surprise since it searches for 1D projections, and any 1D projection of these type of data sets will only reveal unimodal distri-

Table 1. Data set properties along with accuracy and running time results used for the Accuracy benchmark

	size	clusters	dim	LM dim	LMCLUS	ORCLUS	DBSCAN	HPCluster
D_1	3000	3	4	2-3	95% / 0:0:08	80% / 0:0:22	34.6% / 0:0:9	72% / 0:0:51
D_2	3000	3	20	13-17	98.4% / 0:0:33	58.8% / 0:2:18	65.5% / 0:0:36	97.4% / 0:1:39
D_3	30000	4	30	1-4	100% / 0:15:38	64.9% / 1:5:30	100% / 1:31:52	99.3% / 0:1:32
D_4	6000	3	30	4-12	99.9% / 0:9:22	98.3% / 0:8:20	66.5% / 0:3:49	97.1% / 0:0:12
D_5	4000	3	100	2-3	100% / 0:0:20	87.9% / 0:54:30	65.3% / 0:5:24	99% / 0:3:54
D_6	90000	3	10	1-2	99.99% / 0:0:29	100% / 0:29:02	66.7% / 4:58:49	100% / 0:1:23
D_7	5000	4	10	2-6	99.24% / 0:2:05	99.3% / 0:2:41	74.1% / 0:0:54	96% / 0:0:35
D_8	10000	5	50	1-4	99.9% / 0:1:42	63.64% / 1:33:52	100% / 0:17:00	99.2% / 0:3:43
D_9	80000	8	30	2-7	99.9% / 3:12:46	96.9% / 13:30:30	100% / 10:51:15	99.9% / 0:4:57
D_{10}	5000	5	3	1-2	86.5% / 0:0:48	68.2% / 0:0:45	59.6% / 0:0:5	78% / 0:0:33
* D_{11}	1500	3	3	1	98.5% / 0:0:01	99.6% / 0:0:10	42.6% / 0:0:02	33.3% / 0:0:52
* D_{12}	1500	3	3	2	97% / 0:0:02	99% / 0:0:11	33.8% / 0:0:02	33.3% / 0:0:26
* D_{13}	1500	3	7	3	97.7% / 0:0:05	99.1% / 0:0:17	33.9% / 0:0:04	33.3% / 0:0:34
* D_{14}	5000	5	20	4	99.9% / 0:5:46	100% / 0:10:42	21.1% / 0:1:39	20% / 0:1:30
* D_{15}	4000	4	50	3	99% / 0:9:14	100% / 0:25:52	25% / 0:2:34	25% / 0:3:20

butions. However its ability to cluster well the first type of data sets supports the concept of random projections which LMCLUS also implements. In terms of running time, LMCLUS ranked second after HPCluster. The remarkable low running times of HPCluster can be attributed to the fact that it is based on a stochastic procedure which tries a constant number of 1D projections to discover inter-cluster separations, and thus invariant to the size of the data set. Nonetheless LMCLUS runs faster than the other algorithms on seven of the fifteen data sets, and when compared to ORCLUS and DBSCAN only, demonstrates a significant gain in efficiency, especially when applied on large or high dimensional data sets.

4.3 Scalability

We measured the scalability of LMCLUS in terms of size and dimensions. In the first set of tests, we fixed the number of dimensions at ten, and the number of clusters to three, each of which was embedded in a three-dimensional manifold. We then increased the number of points from 1,000 to 1,000,000. In the second set of tests we fixed the number of points, and clusters as before, but increased the number of dimensions from 10 to 120. Fig. 3 is a plot of the running times of LMCLUS in comparison to the other algorithms. The figure shows that in practice, for data sets with a small number of clusters which are embedded in low dimensional manifolds, LMCLUS, like ORCLUS scales linearly with respect to the size of the data set. This can be attributed to the sampling scheme it uses and to the fact that each cluster that is detected is removed from the data set. We note however that as the dimensionality of manifolds increases, performance is likely to degrade. The figure also shows that LMCLUS, like DBSCAN scales linearly with respect to the dimensionality of the data set. Combined together, linearity in both the size and dimensionality of the data set makes LMCLUS one of the fastest algorithms in its class.

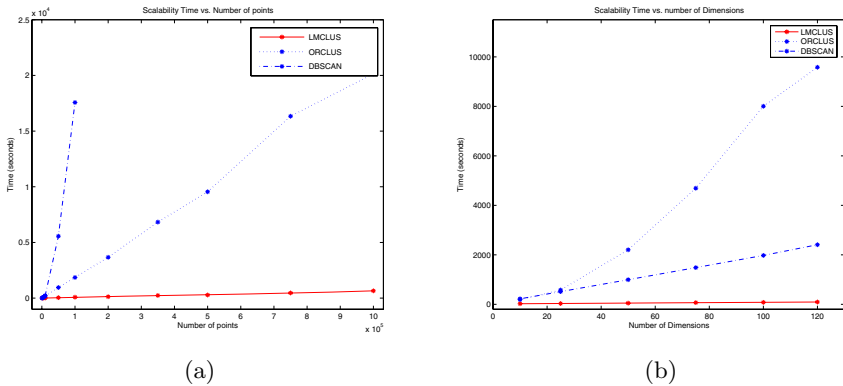


Fig. 3. Scalability, (a) running time vs. data size. (b) running time vs. dimensionality

4.4 Real Data and Applications

Time Series Clustering/Classification. In this experiment we applied LMCLUS on a Time Series data set obtained from the UCI KDD Archive [7] consisting of 600 examples with 60 attributes each, divided into 6 different classes. The donors of this data set claim the this is a good data set to test time series clustering because Euclidean distance measures will not be able to achieve good accuracy. LMCLUS was able to achieve an average of 87% with a high of 89% accuracy. ORCLUS was only able to achieve a high of 50% accuracy, while DBSCAN with extensive tuning of its parameters achieved a high of 68% accuracy, and HPCluster a high of 63.5%.

Handwritten Digit Recognition. The data used in this experiment consists of 3823 handwritten digit bitmaps of 64 attributes each, obtained from the UCI Machine Learning Repository [4]. We divided the data into the even and odd digits, and clustered each separately. LMCLUS was able to achieve an average of 95% and 82% for the even and odd digits respectively, whereas DBSCAN 82% and 58%, ORCLUS 84.7% and 82.9%, and HPCluster 50.3% and 93%.

E3D Point Cloud Segmentation. DARPA’s “Exploitation of 3D Data” identification system must take as input a 3D point cloud of a military target and then compare it to a database of highly detailed 3D CAD models. The first step to accomplish this task usually involves segmenting the targets into their constituent parts. In this experiment we demonstrate LMCLUS’s usefulness as a segmentation procedure. Specifically, LMCLUS was applied on 3D vehicle point cloud CAD models obtained from ALPHATECH Inc., as these provide a similar level of complexity, to that of military vehicles. The applicability of LMCLUS to this problem results from the fact that the surfaces constituting the vehicles closely correspond to 2D linear manifolds embedded in a 3D space. The results of this experiment applied on two different vehicles are depicted in Fig. 4. These results clearly demonstrate LMCLUS’s ability to identify with high precision 2D linear manifolds.

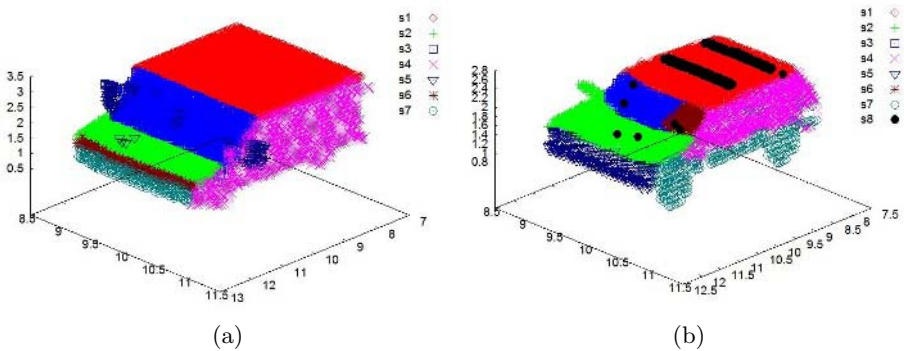


Fig. 4. (a) 2D view of a segmented Aeromate delivery van 3D point cloud (b) 2D view of a segmented Ford Explorer 3D point cloud

5 Conclusion

In this paper we explored the concept of linear manifold clustering in high dimensional spaces. We proposed a new cluster model and showed its relevance to subspace and correlation clustering. Based on this model we presented our algorithm LMCLUS, and demonstrated its superiority over methods such as ORCLUS, DBSCAN, and HPCluster for linear manifold clustering. In addition we presented a successful application of our algorithm to the problem of 3D point cloud segmentation. In the future we plan to investigate the applicability of linear manifold clustering to microarray gene expression clustering, and its usefulness as a tool for modeling high dimensional probability distributions.

References

1. Charu C. Aggarwal, Joel L. Wolf, Philip S. Yu, Cecilia Procopiuc, and Jong Soo Park. Fast algorithms for projected clustering. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 61–72. ACM Press, 1999.
2. Charu C. Aggarwal and Philip S. Yu. Finding generalized projected clusters in high dimensional spaces. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 70–81, 2000.
3. Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM SIGMOD international conference on Management of data*, pages 94–105, 1998.
4. C.L. Blake and C.J. Merz. UCI Repository of machine learning databases. <http://www.ics.uci.edu/mllearn/MLRepository.html> (1998).
5. S. Dasgupta. Learning mixtures of gaussians. In *In Proc. of the 40th Ann. IEEE Symp. on Foundations of Computer Science*, 1999.
6. R.M. Haralick, J.E. Rome, and Alexei Miasnikov. A hierarchical projection pursuit clustering algorithm. In *17th International Conference on Pattern Recognition (ICPR)*, 2004.
7. S. Hettich and S. D. Bay. The UCI KDD archive. <http://kdd.ics.uci.edu> (1999).
8. Ester M. and Kriegel H.P., Sander J., and Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96)*, pages 226–231, Portland, OR, 1996.
9. J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern Recogn.*, 19(1):41–47, 1986.
10. J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. Berkeley, University of California Press, 1967.
11. H. Nagesh, S. Goil, and A. Choudhary. Mafia: Efficient and scalable subspace clustering for very large data sets, 1999.

Universal Clustering with Regularization in Probabilistic Space^{*}

Vladimir Nikulin¹ and Alex J. Smola²

¹ Computer Science Laboratory, Australian National University,
Canberra, ACT 0200, Australia
vladimir.nikulin@anu.edu.au

² NICTA, Canberra, ACT 0200, Australia
alex.smola@nicta.com.au

Abstract. We propose universal clustering in line with the concepts of universal estimation. In order to illustrate above model we introduce family of power loss functions in probabilistic space which is marginally linked to the Kullback-Leibler divergence. Above model proved to be effective in application to the synthetic data. Also, we consider large web-traffic dataset. The aim of the experiment is to explain and understand the way people interact with web sites.

The paper proposes special regularization in order to ensure consistency of the corresponding clustering model.

1 Introduction

Clustering algorithms group empirical data according to the given criteria into several clusters with relatively stable and uniform statistical characteristics.

In this paper we consider prototype-based or distance-based clustering model. The corresponding solution may be effectively approximated using k -means algorithm within *Clustering-Minimization (CM)* framework [1] which may regarded as an analog of the *EM (Expectation-Maximization)* framework for soft clustering or segmentation.

Recently, the Divisive Information-Theoretic Feature Clustering algorithm in probabilistic space \mathcal{P}^m was proposed by [2]. It provides an attractive approach based on the Kullback-Leibler divergence. According to [3], the probabilistic model can be extremely useful in many applications including information retrieval and filtering, natural language processing, machine learning from text and in related areas.

As it is outlined in [4] and [5], in practice, however, an exact form of a loss function is difficult to specify. Hence, it is important to study the domination criterion simultaneously under a class of loss functions. Respectively, we introduce

^{*} This work was supported by the grants of the Australian Research Council. National ICT Australia is funded through the Australian Government initiative.

the family of power loss functions in probabilistic space with KL -divergence as a marginal limit.

Pollard [6] demonstrated that distance-based clustering model in \mathbb{R}^m is consistent under some conditions of general nature. Further, [7] introduced definition of trimmed or robustified k -means and proved consistency of the corresponding model, [8] extended result of [6] to the clustering model with Projection Pursuit which is regarded as a common technique in data analysis with such main advantage as to reduce dimensionality of the data in order to improve its visualization.

We propose definition of α -regularized KL -divergence. On the one hand, in most cases, the corresponding α -regularized clustering model may be made close to the original model with KL -divergence according to the given requirements. On the other hand, α -regularized model will be always consistent.

2 Prototype-Based Approach

Suppose that $\mathbf{X} := \{x_1, \dots, x_n\}$ is a sample of i.i.d. observations drawn from probability space $(\mathcal{X}, \mathcal{A}, \mathbb{P})$ where probability measure \mathbb{P} is assumed to be unknown.

We denote by $\mathcal{Q} \in \mathcal{X}^k$ a codebook as a set of *prototypes* $q(c)$ indexed by the code $c = 1..k$ where k is a *clustering size*.

Following [6] we estimate actual distortion error

$$\mathfrak{R}^{(k)}[\mathcal{Q}, \Phi] := \mathbf{E} \Phi(x|\mathcal{Q})$$

by the empirical error

$$\mathfrak{R}_{\text{emp}}^{(k)}[\mathcal{Q}, \Phi] := \frac{1}{n} \sum_{t=1}^n \Phi(x_t|\mathcal{Q}) \tag{1}$$

where $\Phi(x|\mathcal{Q}) := \Phi(x, q(c(x)))$, $\Phi(\cdot, \cdot)$ is a loss function, and

$$c(x) := \underset{c \in \{1..k\}}{\operatorname{argmin}} \Phi(x, q(c)). \tag{2}$$

Above rule will split the given sample \mathbf{X} into k empirical clusters: $\mathbf{X}_c := \{x_t : c(x_t) = c\}$, $\mathbf{X} = \cup_{c=1}^k \mathbf{X}_c$, $\mathbf{X}_i \cap \mathbf{X}_c = \emptyset, i \neq c$. Similarly, we can define set of k actual clusters $\mathcal{X}_c, c = 1..k$.

Definition 1. We will call $\overline{\mathcal{Q}}$ as an optimal actual codebook if

$$\mathfrak{R}^{(k)}[\overline{\mathcal{Q}}, \Phi] := \inf_{\mathcal{Q} \in \mathcal{X}^k} \mathfrak{R}^{(k)}[\mathcal{Q}, \Phi]. \tag{3}$$

We will call \mathcal{Q}_n as an optimal empirical codebook if

$$\mathfrak{R}_{\text{emp}}^{(k)}[\mathcal{Q}_n, \Phi] := \inf_{\mathcal{Q} \in \mathcal{X}^k} \mathfrak{R}_{\text{emp}}^{(k)}[\mathcal{Q}, \Phi]. \tag{4}$$

Note that an outcome of the k -means algorithm is not necessarily \mathcal{Q}_n as it is defined in (4).

2.1 CM Framework

The algorithm 2.1 represents a typical structure of an algorithm within *CM*-framework.

Algorithm 1. *CM*

- 1: **Clustering:** encode any observation x_t according to the rule (2).
- 2: **Minimization:** re-compute centroids specifically for any particular empirical cluster

$$q(c) := \operatorname{arginf}_{a \in \mathcal{X}} \sum_{x_t \in \mathbf{X}_c} \Phi(x_t, a). \quad (5)$$

- 3: Test: compare previous and current codebooks \mathcal{Q} . Go to the step 1 if convergence test is not fulfilled, alternatively, stop the algorithm.
-

The following Proposition 1, which may be proved similarly to the Theorems 4 and 5 of [2], formulates the most important descending and convergence properties of the *CM*-algorithm.

Proposition 1. *The algorithm 2.1*

- 1) *monotonically decreases the value of the objective function (1);*
- 2) *converges to the local minimum in a finite number of steps if equation (5) has unique solution.*

3 Probabilistic Framework

Let \mathcal{P}^m be the m -dimensional probability simplex or probabilistic space of all m -dimensional probability vectors. Following [2] we assume that the probabilities $p_{it} = P(i|x_t)$, $\sum_{i=1}^m p_{it} = 1$, $t = 1..n$, represent relations between observations x_t and attributes or classes $i = 1..m$, $m \geq 2$. Accordingly, we define the clustering model (\mathcal{P}^m, KL) with *Kullback-Leibler* divergence:

$$KL(\mathbf{v}, \mathbf{u}) := \sum_{i=1}^m v_i \cdot \log \frac{v_i}{u_i} = \langle \mathbf{v}, \log \frac{\mathbf{v}}{\mathbf{u}} \rangle, \mathbf{v}, \mathbf{u} \in \mathcal{P}^m. \quad (6)$$

The following notations will be used below $p(x_t) = \{p_{1t}, \dots, p_{mt}\}$, $q(c) = \{q_{1c}, \dots, q_{mc}\}$.

3.1 Power Loss Functions in Probabilistic Space

Let us consider 2 families of loss functions

$$L\Phi_\gamma(\mathbf{v}, \mathbf{u}) := \sum_{i=1}^m v_i^{1+\gamma} u_i^{-\gamma} - 1, \quad 0 < \gamma < \infty; \quad (7)$$

$$R\Phi_\gamma(\mathbf{v}, \mathbf{u}) := 1 - \sum_{i=1}^m v_i^{1-\gamma} u_i^\gamma, \quad 0 < \gamma < 1. \quad (8)$$

Proposition 2. *The loss functions (7) and (8) are non-negative and equal to 0 if and only if $\mathbf{u} = \mathbf{v}$.*

Above statement may be proved using the method of mathematical induction.

Proposition 3. *Suppose that $\mathbf{v} \neq \mathbf{u}$, $\min\{v_i\} > 0$ and $\min\{u_i\} > 0$. Then, the loss function (7) is convex and strictly increasing as a function of γ .*

Proof. The required result follows from the structure of the corresponding first

$$\frac{\partial L\Phi_\gamma(\mathbf{v}, \mathbf{u})}{\partial \gamma} = \sum_{i=1}^m v_i \log\left(\frac{v_i}{u_i}\right) \left(\frac{v_i}{u_i}\right)^\gamma \quad (9)$$

and second derivatives where the first derivative is strictly positive for $\gamma = 0$ and is strictly increasing for all $\gamma > 0$. \square

Proposition 4. *Suppose that $\mathbf{v} \neq \mathbf{u}$, $\min\{v_i\} > 0$ and $\min\{u_i\} > 0$. Then, the loss function (8) is concave and strictly increasing locally as a function of γ at the point of origin 0:*

$$\exists \varepsilon > 0 : R\Phi_\alpha(\mathbf{v}, \mathbf{u}) < R\Phi_\gamma(\mathbf{v}, \mathbf{u}) \quad \forall \alpha, \gamma : 0 \leq \alpha < \gamma \leq \varepsilon.$$

Proof. The required result follows from the structure of the corresponding derivative

$$\frac{\partial R\Phi_\gamma(\mathbf{v}, \mathbf{u})}{\partial \gamma} = - \sum_{i=1}^m v_i \log\left(\frac{u_i}{v_i}\right) \left(\frac{u_i}{v_i}\right)^\gamma \quad (10)$$

and

$$\frac{\partial^2 R\Phi_\gamma(\mathbf{v}, \mathbf{u})}{\partial \gamma^2} = - \sum v_i \log^2\left(\frac{u_i}{v_i}\right) \left(\frac{u_i}{v_i}\right)^\gamma < 0 \quad (11)$$

where the first derivative is strictly positive for $\gamma = 0$ and is strictly decreasing for all $0 < \gamma \leq 1$. Respectively, $\exists \varepsilon > 0$ so that the first derivative is strictly positive for $0 < \gamma \leq \varepsilon$ as a continuous function of γ . \square

We can compute centroids for the loss functions (7) and (8) in analytical form similar to (12). For example, the following formula represents centroids for (7)

$$q_i(c) \propto {}^{1+\gamma}\sqrt{A_{ic}(\gamma)}, \quad 0 \leq \gamma < \infty, \quad (12)$$

where $A_{ic}(\gamma) = \sum_{x_t \in \mathbf{X}_c} p_{it}^{1+\gamma}$.

Using result of the Proposition 2 we can define a new family of loss functions as an average of (7) and (8)

$$\Phi_\gamma(\mathbf{v}, \mathbf{u}) := \frac{1}{2} (L\Phi_\gamma(\mathbf{v}, \mathbf{u}) + R\Phi_\gamma(\mathbf{v}, \mathbf{u})), \quad 0 < \gamma < 1. \quad (13)$$

The following result demonstrates that the *KL*-divergence may be regarded as a marginal limit in relation to the family of loss functions (13).

Proposition 5. *The family of power loss functions (13) is marginally linked to the KL-divergence: $\lim_{\gamma \rightarrow 0} \frac{\Phi_\gamma(\mathbf{v}, \mathbf{u})}{\gamma} = KL(\mathbf{v}, \mathbf{u})$.*

Proof. The statement of the proposition follows from the structure of the derivative:

$$\frac{\partial \Phi_\gamma(\mathbf{v}, \mathbf{u})}{\partial \gamma} = \frac{1}{2} \sum_{i=1}^m v_i \log \frac{v_i}{u_i} \left[\left(\frac{v_i}{u_i} \right)^\gamma + \left(\frac{u_i}{v_i} \right)^\gamma \right]. \tag{14}$$

In the case if $\gamma = 0$ the right part of (14) equals to the KL-divergence. □

Proposition 6. *Suppose that $\mathbf{v} \neq \mathbf{u}$, $\min \{v_i\} > 0$ and $\min \{u_i\} > 0$. Then, the loss function Φ_γ defined in (13) is strictly increasing locally as a function of γ at the point 0 $\exists \varepsilon > 0 : \Phi_\alpha(\mathbf{v}, \mathbf{u}) < \Phi_\gamma(\mathbf{v}, \mathbf{u}) \quad \forall \alpha, \gamma : 0 \leq \alpha < \gamma \leq \varepsilon$.*

Proof follows from above Propositions 3 and 4.

Remark 1. The results of the Propositions 4 and 6 may not necessarily take place for $\varepsilon = 1$, because $KL(\mathbf{u}, \mathbf{v}) \rightarrow \infty$ if $v_1 \rightarrow 0$ and $\min \{u_i\} \geq \delta > 0$. As a consequence, the derivative (9) is limited. At the same time derivative (10) tends to $-\infty$ if $\gamma \rightarrow 1$ (see Figure 1(d)).

Minimizing $\sum_{x_t \in \mathbf{X}_c} \Phi_\gamma(p(x_t), q) = \sum_{i=1}^m (A_{ic}(\gamma)q_i^{-\gamma} - A_{ic}(-\gamma)q_i^\gamma)$ as a function of $q \in \mathcal{P}^m$ we will formulate iterative algorithm for the computation of centroids in the sense of the loss function (13) with fixed value of the parameter $\gamma > 0$

$$q_i(c, j + 1) \propto \sqrt[1+\gamma]{A_{ic}(\gamma) + A_{ic}(-\gamma)q_i^{2\gamma}(c, j)} \tag{15}$$

where j is a sequential number of iteration, initial values of $q(c, 1)$ may be computed using (12).

Remark 2. According to [5], it seems rather natural to investigate the situation where the estimator is the same for every loss from a certain set of loss functions under consideration. In line with Propositions 3, 4 and 6 we can use parameter γ in order to increase differentiation between observations. Comparing clustering results for different input parameters γ we can make assessment of the stability of clustering: the smaller fluctuation of the centroids will indicate the higher quality of clustering (see Figure 1).

3.2 Consistency of the Clustering Model

According to [9], p. 33, it is extremely important to use concepts that describe necessary and sufficient conditions for consistency. This guarantees that the constructed theory is general and cannot be improved from the conceptual point of view.

Definition 2. *We say [9] that the clustering model (\mathcal{X}, Φ) is consistent if*

$$\mathfrak{R}_{\text{emp}}^{(k)}[\mathcal{Q}_n, \Phi] \xrightarrow[n \rightarrow \infty]{} \mathfrak{R}^{(k)}[\overline{\mathcal{Q}}, \Phi] \quad a.s. \tag{16}$$

We say [6] that the clustering model (\mathcal{X}, Φ) is ν -strongly consistent if

$$\nu(\mathcal{Q}_n, \overline{\mathcal{Q}}) \xrightarrow{n \rightarrow \infty} 0 \quad a.s. \quad (17)$$

where ν is a distance in \mathcal{X}^k .

Definition 3. We will call the element $\mathbf{v} \in \mathcal{P}^m$ as 1) an **uniform vector** if $v_i = \frac{1}{m}, i = 1..m$; and 2) as **i -margin** if $v_i = 0$.

Definition 4. We will call $KL_\alpha(\mathbf{v}, \mathbf{u}) := KL(\mathbf{v}_\alpha, \mathbf{u}_\alpha)$ as α -regularized KL -divergence where $\mathbf{v}_\alpha = \alpha\mathbf{v} + (1 - \alpha)\mathbf{v}_0$ and $\mathbf{u}_\alpha = \alpha\mathbf{u} + (1 - \alpha)\mathbf{v}_0$, \mathbf{v}_0 is an uniform vector and $0 < \alpha \leq 1$.

The following result represents an essential generalization of the Lemma 3 [2].

Proposition 7. Centroids $q(c)$ in $(\mathcal{P}^m, KL_\alpha)$ are not dependent on $0 < \alpha \leq 1$ and must be computed using k -means (12).

Corollary 1. Suppose that $q(c) \in \mathcal{Q}_n$ and $q_i(c) = 0$. Then, $P(i|x_t) = 0 \forall x_t \in \mathcal{X}_c$. Suppose that $q(c) \in \overline{\mathcal{Q}}$ and $q_i(c) = 0$. Then, $P(i|x) = 0 \forall x \in \mathcal{X}_c$ a.s.

Theorem 1. Suppose that the clustering size k and parameter $0 < \alpha < 1$ are fixed. Then, the model $(\mathcal{P}^m, KL_\alpha)$ is consistent.

Proof. The required result

$$\mathfrak{R}_{\text{emp}}^{(k)}[\mathcal{Q}_n^{(\alpha)}, KL_\alpha] \xrightarrow{n \rightarrow \infty} \mathfrak{R}^{(k)}[\overline{\mathcal{Q}}^{(\alpha)}, KL_\alpha] \quad a.s.$$

follows from uniform continuity of the $KL_\alpha(\mathbf{v}, \mathbf{u})$ as a function of both arguments if $0 < \alpha < 1$ where $\mathcal{Q}_n^{(\alpha)}$ and $\overline{\mathcal{Q}}^{(\alpha)}$ are optimal empirical and actual codebooks which correspond to KL_α . \square

Corollary 2. Suppose that the optimal actual codebook $\overline{\mathcal{Q}}^{(\alpha)}$ is unique. Then, the model $(\mathcal{P}^m, KL_\alpha)$ is ν -strongly consistent where a distance ν may be defined as $\max_{c=1}^k \min_{j=1}^k KL(q_n(c), \bar{q}(j))$ where $q_n(c) \in \mathcal{Q}_n^{(\alpha)}$ and $\bar{q}(j) \in \overline{\mathcal{Q}}^{(\alpha)}$.

3.3 Extension to the Euclidean Space

Monograph [10], pp. 255-258, discusses characterization of families of distributions for which the Pitman estimator of the location parameter in \mathbb{R} does not depend on the loss function. Generally speaking, for the same distribution function F , the Pitman estimator differs from loss function to loss function. However, if F is a normal distribution function, then it is easy to see that, for quadratic trigonometrical and following below exponential loss functions (18), the Pitman estimator is one and the same, namely, sample mean.

The G -means algorithm [11] which is based on the *Gaussian* fit of the data within particular cluster is relevant here. The G -means algorithm is based on a statistical test for the hypothesis that a subset of data follows a Gaussian distribution. G -means runs k -means with increasing k in a hierarchical fashion until the test accepts the hypothesis that the data assigned to each centroid are Gaussian.

Similar to the Sect. 3.1 we can define model of universal clustering in \mathbb{R}^m with the following family of exponential loss functions: $\Phi_\gamma(\mathbf{v}, \mathbf{u}) := \varphi_\gamma(\mathbf{v} - \mathbf{u})$ where $\mathbf{v}, \mathbf{u} \in \mathbb{R}^m$, and $\gamma \in \mathbb{R}_+^m$ is m -dimensional regulation parameter,

$$\varphi_\gamma(\mathbf{v}) := \sum_{i=1}^m \cosh(\gamma_i \cdot v_i) - m, \tag{18}$$

and corresponding centroids:

$$q_i^{(\gamma)}(c) = \frac{1}{2\gamma_i} \log \frac{\sum_{\mathbf{x}_t \in \mathbf{X}_c} e^{\gamma_i x_{ti}}}{\sum_{\mathbf{x}_t \in \mathbf{X}_c} e^{-\gamma_i x_{ti}}}$$

which represent a unique k -means solution for the loss function (18).

4 Experiments

The sample of the 3D-probability data, which is displayed in the Figures 1 was generated using the following procedure.

Firstly, the cluster code c was drawn randomly according to the probabilities p , see Table 1, using standard uniform random variable. Secondly, we used the multinomial logit model in order to generate coordinates of the 3D-probability data: $v_i \propto \exp\{b_{ci} + e_c r\}$, $\sum_{i=1}^3 v_i = 1$, where r is a standard normal random variable.

By definition, the family of power loss functions (13) is marginally linked to the KL -divergence if $\gamma \rightarrow 0$. By the increase of γ we will increase the power of diversification. Respectively, any centroid, which corresponds to a non significant empirical cluster will move around. Figure 1 illustrates that centroids of the

Table 1. Simulation coefficients for the 3D-synthetic data, see Figure 1

Cluster	Coefficients				Probabilities
c	b_1	b_2	b_3	e	p
1	1	-1	-1	0.5	0.15
2	-1	1	-1	0.5	0.15
3	-1	-1	1	0.5	0.15
4	-0.4	-0.4	-0.8	0.4	0.25
5	-0.4	-1.9	-0.4	0.3	0.15
6	-1.9	-0.4	-0.4	0.3	0.15

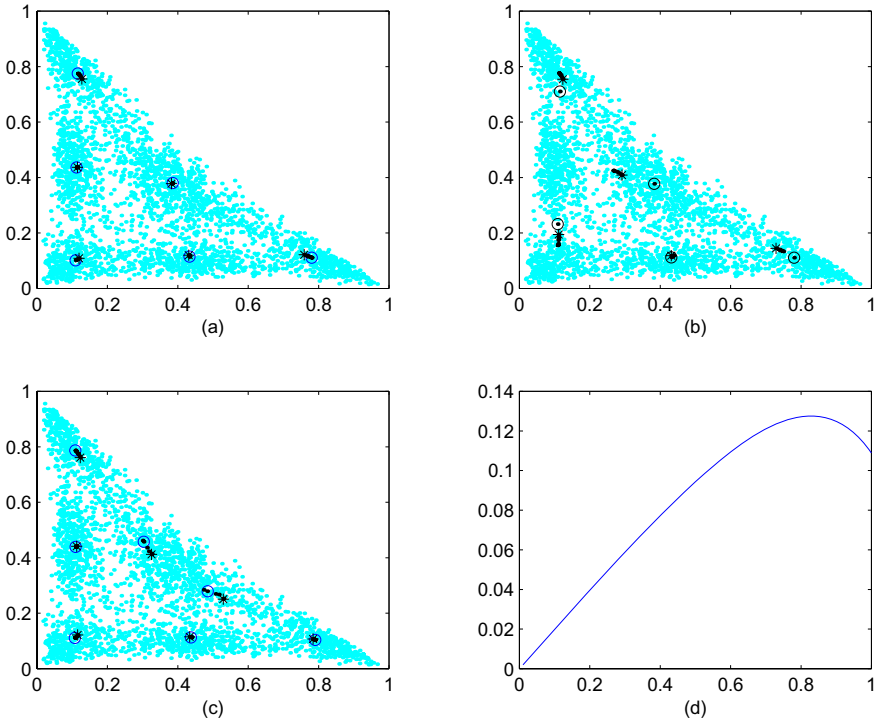


Fig. 1. 3D-synthetic data, $n = 3000$ with 6 clusters, (a): $k=6$: random selection of the cluster seeds; centroids were re-computed using loss function (13) with $\gamma = 0.09 + 0.13 \cdot (i - 1), i = 1..8$; symbol \odot marks centroids which corresponds to $\gamma = 0.09$; * marks centroids which corresponds to $\gamma = 1.0$, other centroids are marked by bold black dots ; (b): $k=5$; (c): $k=7$; (d): loss (13) as a function of γ where $m = 10, u_i = \frac{1}{m}, i = 1..m, v_1 = \varepsilon, v_i = \frac{1-\varepsilon}{m-1}, i = 2..m, \varepsilon = 0.001$

“strong” empirical clusters are stable as a consequence of correct selection of the number of clusters $k = 6$.

The second experiment was conducted using a large Web navigation **msnbc** dataset. This dataset comes from Internet Information Server **msn.com** for the entire day of *September, 28, 1999* [12]. The dataset [13] includes $n = 989818$ sequences of events with lengths ranging from 1 to 12000.

Each sequence in the dataset corresponds to page views of a user during that twenty-four hour period. Each event in the sequence corresponds to a user’s request for a page. In total, there are 4698794 events.

The page categories were developed prior to investigation. There are $m = 17$ particular web categories. The number of pages per category ranges from 10 to 5000.

Analysis of the **msnbc** data had revealed the following general properties: 1) users have tendency to stay within particular category; 2) transitions from one category to another are relatively rare.

Algorithm 2. (Universal Clustering)

- 1: Order number of clusters k , and select randomly initial codebook with k probability vectors which will be used for all $\tau \geq 2$ runs of the *CM* algorithm in the next step.
- 2: Run *CM*-algorithm using loss function (13) with $\gamma = \gamma_0 + (j-1) \cdot \delta, j = 1.. \tau$, where $0 < \gamma_0 < 1$ and $0 < \delta \leq \frac{1-\gamma_0}{\tau-1}$. As an outcome we obtain a set of $k \cdot \tau$ probability vectors $\{\tilde{q}(j, c), j = 1.. \tau, c = 1..k\}$.
- 3: Compute maximum distance between first and other codebooks

$$D := C \cdot \max_{c=1..k} \max_{j=2.. \tau} KL(\tilde{q}(1, c), \tilde{q}(j, c)) \quad (19)$$

where $C > 0$ is a constant.

Respectively, we considered an ultimate simplification of the model by ignoring 1) dependencies between subsequent events and 2) length of the sequence of events for any particular user. As a result, we reduced the given variable-length data to the fixed length data where any user is represented by the m -dimensional probability vector of the frequencies of m categories.

The aim of this experiment is to explain and understand the way people interact with web sites, explore human behavior within internet environment. Briefly, we observed that the table of centroids in the case of $k = 8$ demonstrates clearly user's preferences. Detailed numerical and graphical illustrations may be found in [1].

Also, the paper [1] introduced clustering regularisation based on the balanced complex of two conditions: 1) significance of any particular cluster; 2) difference between any 2 clusters. Subject to some input regulation parameters the corresponding system detected the interval $34 \leq k \leq 47$ as the most likely range for the number of significant clusters in **msnbc**. Another solution for the same task may be found using principles of universal clustering.

Table 2. 3D-probabilistic synthetic data: determination of the clustering size k where D is defined in (19), used parameters: $\gamma_0 = 0.002, \delta = 0.01, \tau = 20, C = 1000$

k:	3	4	5	6	7	8	9
D:	0.6478	0.0263	0.0045	0.0011	0.8535	0.9264	2.7150
k:	10	11	12	13	14	15	16
D:	0.8041	1.9056	0.1474	0.3063	0.9377	5.0651	12.1121

A Pentium 4, 2.8GHz, 512MB RAM, computer was used for the computations. The overall complexity of a CM cycle is $O(k \cdot n \cdot m)$. The computer conducted computations according to the special program written in C. The computation time for one CM cycle in the case of 51 clusters was 110 seconds.

5 Concluding Remarks

Experiments on the real and synthetic data had confirmed fast convergence of the CM -algorithm [1]. Unfortunately, the final results of the CM -algorithm depend essentially on initial settings, because the algorithm may be trapped in local minimum. In this regard, the proposed in the Section 3.2 α -regularization is significant because it will guarantee consistency of the corresponding clustering model. On the other hand, the proposed in the paper universal clustering represents a promising direction. We can make an assessment of quality of clustering using set of codebooks as a function of regulation parameter. The quality function may be computed as a decreasing function of the fluctuation of codebooks.

Acknowledgments. We are grateful to Peter Hall for the consideration and very valuable support. Our thanks go also to anonymous referees for the helpful comments and suggestions.

References

- [1] Nikulin, V., Smola, A.: Parametric model-based clustering. In Dasarathy, B., ed.: Data Mining, Intrusion Detection, Information Assurance, and Data Network Security, 28-29 March 2005, Orlando, Florida, USA. Volume 5812., SPIE (2005) 190–201
- [2] Dhillon, I., Mallela, S., Kumar, R.: Divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research* **3** (2003) 1265–1287
- [3] Cohn, D., Hofmann, T.: The missing link - a probabilistic model of document content and hypertext connectivity. In: 13th Conference on Neural Information Processing Systems. (2001)
- [4] Hwang, J.T.: Universal domination and stochastic domination: Estimation simultaneously under a broad class of loss functions. *The Annals of Statistics* **13** (1985) 295–314
- [5] Rukhin, A.: Universal Bayes estimators. *The Annals of Statistics* **6** (1978) 1345–1351
- [6] Pollard, D.: Strong consistency of k-means clustering. *The Annals of Statistics* **10** (1981) 135–140
- [7] Cuesta-Albertos, J., Gordaliza, A., Matran, C.: Trimmed k-means: an attempt to robustify quantizers. *The Annals of Statistics* **25** (1997) 553–576
- [8] Stute, W., Zhu, L.: Asymptotics of k-means clustering based on projection pursuit. *Sankhya* **57** (1995) 462–471
- [9] Vapnik, V.: *The Nature of Statistical Learning Theory*. Springer (1995)

- [10] Kagan, A., Linnik, Y., Rao, C.: Characterization Problems in Mathematical Statistics. John Wiley & Sons (1973)
- [11] Hamerly, G., Elkan, C.: Learning the k in k-means. In: 16th Conference on Neural Information Processing Systems. (2003)
- [12] Cadez, I., Heckerman, D., Meek, C., Smyth, P., White, S.: Model-based clustering and visualization of navigation patterns on a web site. *Data Mining and Knowledge Discovery* **7** (2003) 399–424
- [13] Msnbc: msnbc.com anonymous web data. In: UCI Knowledge Discovery in Databases Archive: <http://kdd.ics.uci.edu/summary.data.type.html>. (1999)

Acquisition of Concept Descriptions by Conceptual Clustering

Silke Jänichen and Petra Perner

Institute of Computer Vision and applied Computer Sciences,
IBaI, Körnerstr. 10, 04107 Leipzig
ibaiperner@aol.com,
www.ibai-institut.de

Abstract. Case-based object recognition requires a general case of the object that should be detected. Real world applications such as the recognition of biological objects in images cannot be solved by one general case. A case-base is necessary to handle the great natural variations in the appearance of these objects. In this paper we will present how to learn a hierarchical case base of general cases. We present our conceptual clustering algorithm to learn groups of similar cases from a set of acquired structural cases. Due to its concept description it explicitly supplies for each cluster a generalized case and a measure for the degree of its generalization. The resulting hierarchical case base is used for applications in the field of case-based object recognition.

Keywords: Case Mining, Case-Based Object Recognition, Cluster Analysis.

1 Introduction

In case-based object recognition a group of similar objects is represented by a generalized case for the purpose of efficient matching. If this representative case is not known *a-priori* it must be learnt from real examples. There arise special problems if the objects of interest have a great variation so they can not be generalized by one single case. A case base is necessary which describes the different appearances of the objects. But then it is also not known in advance how many cases are necessary to detect all objects with a sufficiently high accuracy.

Clustering techniques can be used to mine for groups of similar cases in a set of acquired cases. For each group it is possible to determine a generalized case to represent this group. Because we do not know the number of cases in advance we will use hierarchical cluster analysis method to learn a hierarchy of increasing generalized cases. Applying a hierarchical instead of a flat case-base for case-based object recognition might speed up the recognition process especially in CBR applications with very large case bases.

When learning a representative case of a cluster this case should average over all cases in this cluster by generalizing common properties of the instances. We offer two different approaches to calculate such a representative. While the first one is to learn an artificial case that is positioned in the centroid, the second one selects that case out of a cluster which has the minimum distance to all other cases in this cluster.

It is also important to know the permissible dissimilarity from this generalized case. The degree of generalization of the cases decreases from the top to the bottom of the hierarchical case base and has to be taken into account in the matching process. The more groups are established in a hierarchy level the less generalized these representatives will be. When matching those cases for object recognition the similarity measure has to be set according to the degree of its abstraction. The less generalized the cases are, the higher the required similarity for the matched objects.

We will present in this paper our study on learning generalized cases. First we review related work on clustering in Section 2 and describe the material used for our study in Section 3. After having reviewed some agglomerative clustering methods in Section 4 we describe our novel algorithm for learning general cases in Section 5. The description of the calculation of cluster representatives is given in Section 6. We discuss experimental results in Section 7 and, finally, give conclusions in Section 8.

2 Related Work

Cluster analysis [1], [2] is used to mine for groups of similar observations in a set of unordered observations. In conclusion, similar cases should belong to the same cluster for strong internal compactness and dissimilar cases should belong to different clusters for a maximum of external separation.

There are plenty of different clustering algorithms [3], [4] and which one is best suited depends on the dataset and on the special properties and aims coupled with the cluster analysis. One main difference between several clustering algorithms is the resulting organization of the instances. Clustering algorithms can be distinguished into overlapping, unsharp, and disjunctive. While overlapping clustering algorithms allow that one case is located in one or more clusters, unsharp clustering algorithms assign to each case membership values related to all clusters. Disjunctive clustering algorithms are best suited for our application because every case has to be assigned to exactly one cluster.

Another main criterion concerning the choice of a clustering method is if the number of resulting groups is known. If the number of clusters is known *a-priori* partitioning clustering [5], [6] can be used, where an initial partition of the cases becomes optimized. If it is unknown or impossible to determine the number of clusters in advance it might be better to create a sequence of partitions using hierarchical clustering methods.

A hierarchical clustering method [1], [4] divides the set of all input cases into a sequence of disjunctive partitions. They can be distinguished between agglomerative and divisive methods. Initially, in the agglomerative methods each case is hosted in its separate cluster. With increasing distances the clusters become merged in cluster that are more general until finally all cases are hosted in the same cluster. The opposite is given in the divisive methods, where initially all cases are hosted in one cluster and were splitted until all cases form their own cluster. The main drawback of these algorithms is that once a cluster has been formed there is no way to redesign this cluster if necessary when other examples have been seen.

Another main problem with these conventional clustering algorithms is that it is only possible to draw conclusions about the composition of the clusters. They do not


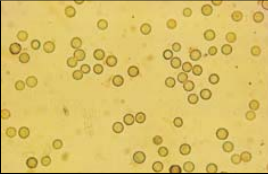
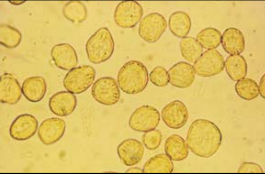
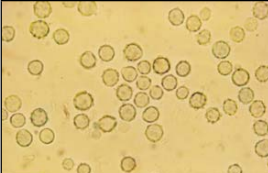
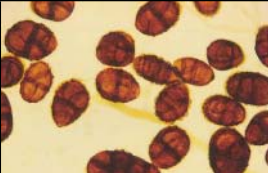

explain why a cluster was established and they supply no real indication about the quality of single partitions. To determine the partition with the optimal number of clusters different cluster validity indices [7], [8] can be used to prove the quality of single partitions. However these indices have to be calculated in an off-line phase after the clustering has been done. Besides conventionally clustering methods supply no precise description about the clusters. One has to calculate this manually for each cluster in a post-processing step which is not sufficient for our purpose.

Alternatively, different conceptual clustering algorithms [9], [10], [11] were developed. They establish clusters with a utility function, which can be built on a probabilistic concept [9], [11], or a similarity based concept [10]. On the basis of this function they explain why a set of cases confirm a cluster and automatically supply a comprehensive description of the established concepts. Their concept forming strategy is more flexible than the one of the conventional hierarchical clustering algorithms.

3 Material

In our application we are studying the shapes of six different airborne fungal spores. Table 1 shows one of the images for each analyzed fungal strain. The objects have a great variance in the appearance so that it is impossible to represent their shape by only one case. But for the purpose that these object shapes should be effectively detected in new images, it is indispensable to generalize the shapes.

Table 1. Images of six different fungal strains

		
Alternaria Alternata	Aspergillus Niger	Rhizopus Stolonifer
		
Scopulariopsis Brevicaulis	Ulocladium Botrytis	Wallenia Sebi

From the real images we acquired a set of shapes for each species. These shapes were pair-wise aligned to obtain a measure of distance between them. A detailed description of our shape acquisition and alignment process was presented in [12]. The alignment of every possible pair of shapes leads us to $N \times N$ distance measures between N acquired cases. These distances can be collected in a matrix where each

row and each column corresponds to a shape instance. This square symmetric distance matrix will be used as input for the hierarchical cluster analysis.

4 Agglomerative Clustering Methods

There are plenty of different agglomerative clustering methods. Each method has its special characteristic and should be used in compliance with the aims of the application, e.g. detection of outlier. We will analyze how they can be used for our problem of learning groups of similar cases and group representatives with its concept description.

Usually in agglomerative clustering methods the resulting sequence of partitions is graphically represented by a dendrogram (see Fig. 1). The set of all input cases is shown on the left side. In the initial partition each case forms its own cluster. They become merged with increasing distances from left to right until all cases are combined in only one cluster. The distance where two clusters become one cluster for the first time depends on the particular clustering method. This distance is called cophenetic proximity measure and is drawn on the abscissa of the dendrogram. Note that this proximity measure is not equal to the pair-wise dissimilarity measure. However the aim while calculating the cophenetic proximity measure is that the real proximity relation between the objects should not be distorted.

To obtain the partition of one level in the hierarchy the dendrogram has to be cut at some distance. The cut-point drawn in Fig. 1 splits the input cases into three clusters.

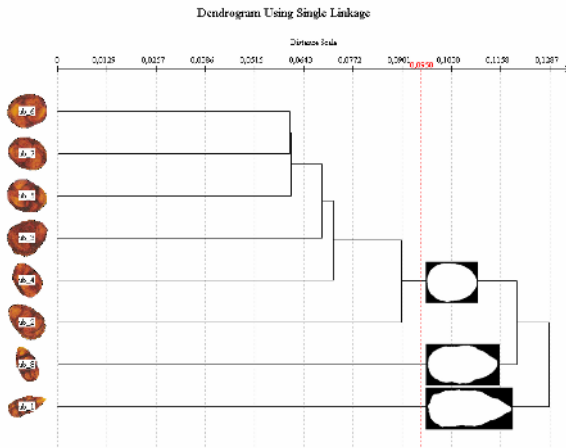


Fig. 1. Dendrogram of eight instances of strain *Ulocladium Botrytis* using single linkage with generalized cases calculated at distance of 0.0950

Since the clusters are merged together on specific converted distances, every method has an own ultra-metric [2]. In the single linkage method two clusters become merged to a new cluster at their minimum distance, the smallest distance between a

case in one cluster and its nearest neighbor in the second cluster. Graphically this can be interpreted as the shortest link between two clusters. The length of this link is the cophenetic proximity measure in the dendrogram where the two clusters become united. It usually leads to long, elongated clusters in the formation of chains which is disadvantageous in most applications. Note that only one single case out of the cluster establishes the link to the second cluster. This means the other cases have no influence on the cophenetic proximity measure. The calculated pair-wise distances between two cases within a cluster might be much greater than the cophenetic proximity measure at this hierarchy level.

By contrast the complete linkage method merges two partitions at their maximum distance, the greatest of all pair-wise distances between a case in the first and a case in the second cluster. If this method has merged the two clusters at some cophenetic proximity, it is guaranteed that every pair-wise distance between two arbitrary cases within the new cluster is smaller than or equal to this measure. Complete linkage usually establishes homogeneous, spherical clusters. Outlier cases stay unrecognized. This might be disadvantageous, because a single outlier might prevent the merging of two groups, although all other cases are very similar. But, the cophenetic proximity measures obtained with complete linkage give a first impression about the expansions of the clusters. An agglomerative clustering method which is a midway between these two extreme methods is average linkage: Two clusters are merged at the average distance of all pair-wise distances between comprised cases. In addition it is possible to weight each cluster according to the number of hosted cases. The average linkage methods establish spherical clusters while outlier cases stay unrecognized for a long time.

In the centroid method the cophenetic proximity between two clusters is the distance between their centroids. The weighting included in the centroid method emphasizes clusters which consist of many cases while small sized clusters tend to get lost [13]. Therefore, in the median method, the weights of the comprised clusters are not included when calculating the centroid of the new cluster. These methods are most suited for our purpose because we are interested in determining a representative case in the cluster centroids. But they also give no real indication about the degree of generalization of the two clusters at a hierarchy level. If necessary we have to calculate this measure in an off-line phase.

In summary it can be said that the agglomerative hierarchical clustering methods give a good impression about the organization of the underlying dataset. However, these algorithms only produce a sequence of partitions but give no further indication about why this cluster was established. Thus all other information concerning a more detailed description of a cluster, e.g. cluster mean, inner-cluster-variance, have to be calculated manually. This fact is a main drawback in all applications where the number of classes is not known in advance. The agglomerative clustering methods are simple but also rigid and inflexible. They offer merging as the only possibility to incorporate a case into a hierarchy. If a case is merged once it is impossible to separate it or to change the cluster again. If it turns out later that a classification was wrong, this is irreversible. Besides that these clustering methods can not be used for incremental learning.

5 Our Conceptual Clustering Algorithm

Conceptual clustering is a type of flexible learning the hierarchy by observations. The partitioning of the cases is controlled by a category utility function [1]. Conceptual clustering algorithms can be distinguished by the type of this utility function which can be based on a probabilistic [9], [11] or a similarity concept [10]. Our conceptual clustering algorithm presented here is based on similarities, because we do not consider logical but numerical concepts. The algorithm works directly with structural objects. In our study this is a set of acquired cases, each comprised by an ordered array of contour points. In contrast to agglomerative clustering methods where the distance matrix is used as input it is not necessary to calculate pair-wise distances in advance.

In addition to merging cases our algorithm allows incorporating new cases into existing nodes, opening new nodes, and splitting of existing nodes at every position in the hierarchy. Each new case is successively incorporated, so the algorithm dynamically fits the hierarchy to the data. The resulting sequence of partitions is represented by a directed graph (concept hierarchy) where the root node contains the complete set of input cases and each terminal node represents an individual case.

Initially the concept hierarchy only consists of an empty root node. The algorithm implements a top-down method. A new case is placed into the actual concept hierarchy level by level beginning with the root node until a terminal node is reached. In each hierarchy level one of these four different kinds of operations is performed:

- The new case is incorporated into an existing child node,
- A new empty child node is created where the new case is incorporated,
- Two existing nodes are merged to form a single node where the new case is incorporated, and
- An existing node is splitted into its child nodes.

The new case is tentatively placed into the next hierarchy level by applying all of these operations. Finally that operation is performed which gives the best score of the partition according to the evaluation criteria. A proper evaluation function prefers compact and well separated clusters. These are clusters with small inner-cluster variances and high inter-class variances. Thus we calculate the score of a partition by

$$SCORE = \frac{1}{m} \sum_{i=1}^m p_i (SB_i - SW_i), \quad (1)$$

where m is the number of clusters in this partition, p_i is the relative frequency of the i -th cluster, SB_i is the inter-cluster variance and SW_i is the inner-cluster variance of the i -th cluster. The normalization according to m is necessary to compare partitions of different size. The relative frequency p_i of the i -th cluster is

$$p_i = \frac{n_i}{n}, \quad (2)$$

where n_i is the number of cases in the i -th cluster and n is the number of cases in the parent node. The output of our algorithm for applying the eight exemplary shape

cases of strain *Ulocladium Botrytis* is shown in Fig. 3. On top level the root node is shown which comprises the set of all input cases. Successively the tree is partitioned into nodes until each input case forms its one cluster.

We introduced a pruning criterion into the algorithm which can be used optionally. It says that the clusters in one partition are removed if the sum of their inner-cluster-variances is zero. Fig. 2 shows the complete, un-pruned concept hierarchy, where a new case was incorporated supplementary. The darker nodes were those clusters which had to be updated because the new case was incorporated into them. The white nodes in the hierarchy are clusters which were not attached.

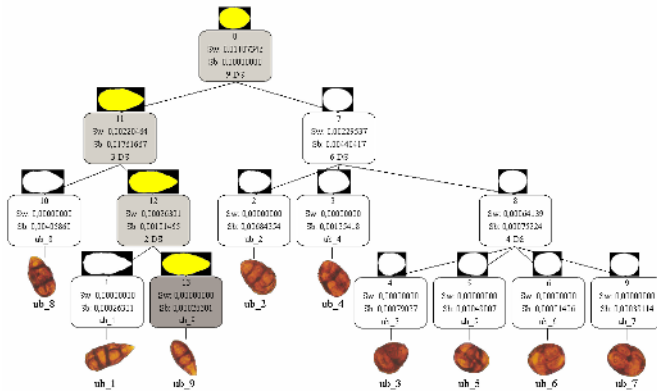


Fig. 2. Complete, not-pruned concept hierarchy after incrementally incorporating a new case. The *darker* nodes are those clusters which are modified because the new case was inserted. Their new representative cases are depicted as *yellow* shapes. The *white* nodes were not attached while the hierarchy was modified to fit the new data

The main advantage of our conceptual clustering algorithm is that it brings along a concept description. Thus, in comparison to agglomerative clustering methods it is easy to understand why a set of cases forms a cluster. The algorithm calculates the inner-cluster-variances direct on the cases within this cluster or rather on their contour points instead of using a given distance matrix. During the clustering process the representative case, and also the variances and maximum distances in relation to this representative case are calculated since they are part of the concept description. The algorithm is of incrementally fashion because it is possible to incorporate new cases into the existing learnt hierarchy.

6 Calculation of General Cases

The representative case of a cluster is a more general representation of all cases hosted in this cluster. Since this case should average over all cases in that cluster, a good case might be positioned in the centroid of the cluster. In our conceptual clustering algorithm the concept description is based on the inner-cluster-variance. The inner-cluster-variance of a cluster X is calculated by

$$SW_X = \frac{1}{n_x} \sum_{i=1}^{n_x} d(x_i, \bar{\mu}_X)^2, \quad (3)$$

where $\bar{\mu}_X$ is the centroid and n_x is the number of cases in the cluster X . Thus, a direct output of the clustering process is the calculation of the cluster centroids.

In our application the cluster centroid is an artificial mean shape defined by an ordered set of points. To calculate this shape it is necessary to determine corresponding points [12] between the shapes in the cluster. For each set of corresponding points between all shapes in one cluster we calculate its centroid. The centroid of a set of n_S corresponding 2D-points $s_i(x, y)$, $i = 1, 2, \dots, n_S$ is given by

$$\bar{\mu}_S(x, y) = \frac{1}{n_S} \sum_{i=1}^{n_S} s_i(x, y). \quad (4)$$

All calculated mean points are set as points on the contour of the representative shape of this cluster. This results in an artificial mean shape case positioned in the centroid of the cluster.

A second approach is to select the medoid as a natural representative case for a cluster. The medoid x_{medoid} of a cluster X is the shape case which is positioned closest to the cluster centroid. It is the case which has the minimum distance to all other cases in the cluster

$$\bar{\mu}_X = x_{medoid} = \min_{x \in X} \sum_{i=1}^{n_x} d(x_i, x). \quad (5)$$

In addition to the representative of a cluster we are interested in leaning the maximum permissible distance from this generalized case. The maximum permissible distance D_X to the representative case is

$$D_X = \max_{x \in X} d(x, \bar{\mu}_X). \quad (6)$$

When matching objects with a hierarchical case-base of increasing specialized cases it is important to know the degree of generalization for each case. This measure will be used as threshold for the similarity score while matching.

7 Experimental Results

Our conceptual clustering algorithm was directly applied to the set of shape cases instead of the matrix of pair-wise distances between those cases. The pruned version of the resulting hierarchy for eight exemplary cases is shown in Fig. 3. The established groups appear useful and logical. If we compare this hierarchy to the outputs of the agglomerative clustering algorithms it is very similar to the median method, which is based on the distances between un-weighted cluster centroids. The outputs are similar but the main difference is how these results were obtained.

In comparison to the agglomerative methods our conceptual clustering algorithm is incremental and more flexible. If during the process it turns out that a classification was wrong, it is still possible to split or merge a formed cluster afterwards. If a new case is incorporated into the concept hierarchy, the algorithm dynamically fit the hierarchy to the new data. It has linear time complexity $O(N)$. By contrast the agglomerative clustering methods have to calculate the complete hierarchy again if a new case should be incorporated supplementary. Thus, conceptual clustering is better suited for huge databases and all applications where it is necessary to adapt the hierarchy by learning new cases over time.

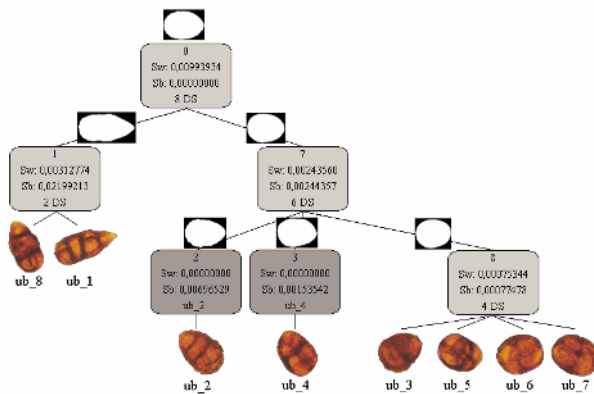


Fig. 3. The pruned version of the concept hierarchy resulting from the eight instances of strain *Ulocladium Botrytis* is shown. On the top of each node the generalized representative of this cluster is shown

Our algorithm brings the right concept description for our purpose of learning case groups and generalized cases. The calculated general cases represent the clusters and are stored into the case base. The measures inner-cluster variance, inter-cluster variance, and maximum permissible distance to the cluster centroid help us to understand on what hierarchy level we should stop to generalize the cases so that we can achieve good results during the matching process.

8 Conclusion

We have described how to learn a hierarchical case base of general cases from a set of acquired cases. It has shown that classical hierarchical clustering methods give a good impression about the organization of the cases but fail if further information is necessary. Our presented conceptual clustering algorithm is directly working on the set of structural cases, while the resulting hierarchy is similar to those of classical hierarchical clustering methods.

We have also shown that our algorithm is more flexible since the establishing of the hierarchy is not only based on merging, but it is also possible to split, incorporate,

and create cluster. In addition to that it allows incremental incorporation of new cases while the hierarchy is only adapted to fit the new data. Due to its concept description our conceptual clustering algorithm supplies for each cluster a generalized case and a measure for the degree of its generalization. This output in form of a hierarchical case base with decreasingly generalized cases is the basis for efficient application in case-based object recognition.

Acknowledgement

This project is sponsored by the European Commission within the project “Multimedia Understanding through Semantics, Computation, and Learning” No. 507752.

References

1. P. Perner, *Data Mining on Multimedia Data*, Springer Verlag Berlin, 1998.
2. H.J. Mucha, *Clusteranalyse mit Mikrocomputern*, Akademie Verlag, Berlin, 1992.
3. A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, 1988.
4. E. Rasmussen, *Clustering Algorithms*, In W.B. Frakes and R. Baeza-Yates (Eds), *Information Retrieval*, pp. 419-442, Prentice Hall, 1992.
5. J.B. MacQueen, *Some Methods for classification and Analysis of Multivariate Observations*, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281-297, Berkeley, University of California Press, 1967.
6. S.K. Gupta, K.S. Rao, and V. Bhatnagar, *K-means Clustering Algorithm for Categorical Attributes*. In M.K. Mohania and A. Min Toja (Eds.) *Proceedings of the First International Conference on Data Warehousing and Knowledge Discovery*, pp. 203-208, Springer Verlag, Inc 1676, 1999.
7. J.C. Dunn, *Well separated clusters and optimal fuzzy partitions*, *J. Cybern.* Vol. 4, pp. 95-104, 1974.
8. D.L. Davies and D.W. Bouldin, *A cluster separation measure*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1, No. 2, pp. 224-227, 1979.
9. D. Fisher and P. Langley, *Approaches to conceptual clustering*, *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pp. 691-697, Los Angeles, 1985.
10. P. Perner, *Different Learning Strategies in a Case-Based Reasoning System for Image Interpretation*, In B. Smith and P. Cunningham (Eds.), *Advances in Case-Based Reasoning*, pp. 251-261, Springer Verlag, Inai 1488, 1998.
11. W. Iba and P. Langley, *Unsupervised Learning of Probabilistic Concept Hierarchies*, In G. Paliouras, V. Karkaletsis, & C.D. Spyropoulos (Eds.), *Machine learning and its applications*. Springer Verlag, 2001.
12. P. Perner and S. Jänichen, *Case Acquisition and Case Mining for Case-Based Object Recognition*, In: Peter Funk and Pedro A. Gonzalez Calero (Eds.), *Advances in Case-Based Reasoning*, *Proceedings of the ECCBR 2004*, pp. 616-629, Springer Verlag, 2004.
13. G.N. Lance and W.T. Williams, *A General Theory of Classification Sorting Strategies*, 1. *Hierarchical Systems*, pp. 373-380, *Comp. J.* 9, 1966.

Clustering Large Dynamic Datasets Using Exemplar Points

William Sia and Mihai M. Lazarescu

Department of Computer Science, Curtin University,
GPO Box U1987, Perth 6001, W.A
{sia, lazarescu}@cs.curtin.edu.au

Abstract. In this paper we present a method to cluster large datasets that change over time using incremental learning techniques. The approach is based on the dynamic representation of clusters that involves the use of two sets of *representative* points which are used to capture both the current shape of the cluster as well as the trend and type of change occurring in the data. The processing is done in an incremental point by point fashion and combines both data prediction and past history analysis to classify the unlabeled data. We present the results obtained using several datasets and compare the performance with the well known clustering algorithm CURE.

1 Introduction

Clustering is the process of discovering sets of similar objects and grouping them into classes based on similar patterns. Existing clustering algorithms can be grouped into several categories with each category attempting to address a different set of problems. Most current algorithms use *static models*. Hence, they generally have a poor performance when confronted with sets of data that do not fit their choice of static model. This happens in particular when dealing with data that contains clusters of varying sizes, shapes and attributes.

Another significant problem is that many of the current algorithms *do not scale well* when handling large data sets. Furthermore, the majority of the existing algorithms are designed to work with static datasets. The work we present in this paper attempts to address the problems of *scalability* and *static models*. We also introduce a further requirement: that of processing dynamic datasets.

The term *dynamic* means that the data representation changes from time to time. There are two reasons for focusing on dynamic datasets. The first is that many of the existing data mining applications involve datasets which change over time while the second reason is the fact that in many domains extracting information about the trend and the changes that occur in the clusters/classes provides crucial clues on how the information should be interpreted (e.g. medical studies that are used to identify the changes in conditions that can trigger the onset of a particular mental/behavioural problem). Most current algorithms do not reflect or interpret these changes in their computation as they merely produce a snapshot of the data. In this paper, the proposed approach detects the changes that occur and tries to incorporate this information to predict future data. As a result, it is able to represent the data dynamically.

We use a hierarchical approach to group the data that uses a subset of representative points to define the clusters discovered. The processing is done incrementally. However, unlike previous clustering methods, we use a number of machine learning techniques derived from drift tracking techniques which enable our algorithm to handle more effectively the dynamic aspect of the data.

The paper is organised as follows: Section 2 presents an overview of the previous research that is related to our work. Section 3 describes the algorithm while in Section 4 we present the two sets of results as well as a comparison with the CURE algorithm. The conclusions are covered in Section 5.

2 Related Work

Clustering is the process of segmenting data sets into groups of similar objects. It usually consists of two basic steps. The first step is to determine the number of possible clusters. Based on that number of clusters, the program tries to fit the best possible clustering[1]. Fields of study such as statistics, pattern recognition, and machine learning utilize clustering in their applications[2]. Clustering algorithms can be categorized into classes such as hierarchical , partitioning , grid-based and density based methods. In hierarchical clustering, a series of partitions take place before the optimal number of clusters is produced. Hierarchical clustering has the advantages of flexibility regarding the level of granularity. However the termination criteria is quite subjective as it differs from case to case. Moreover, the general approach does not perform any general refinement and improvement to clusters that have been constructed.

BIRCH [3] (Balanced Iterative Reducing and Clustering using Hierarchies) is an example of an agglomerative hierarchical clustering algorithm that allows additional passes to improve the cluster quality. *BIRCH* incrementally and dynamically clusters incoming multi-dimensional metric data points to try to produce the best quality clustering with the available resources. *CURE* [4] is a hierarchical clustering algorithm that integrates the centroid-based and all-point extremes approach when representing a cluster. The number of points to represent a cluster is determined by a constant number c previously selected by the user. By using well scattered points, *CURE* can handle clusters with varying sizes. Moreover, this approach results in a better performance compared to *BIRCH* in discovering non-spherical clusters. *CURE* allows the user to select the parameter α , that is used as a fraction when it tries to shrink the chosen scattered-points toward the centroid. These shrunken new points will reside between the centroid and the outline of the clusters. However, using representative points may not achieve an accurate result when selected points are insufficient to define the cluster representation. The problem is compounded when the shrinking phase eliminates some points that are ideal to represent the cluster. *CHAMELEON* [5] solves this shortcoming by determining the pair of most similar sub-clusters by considering both the inter-connectivity (RI) as well as the closeness (RC) of the clusters. *CHAMELEON* utilizes dynamic modeling in cluster aggregation. The algorithm works by finding the K -nearest neighbors for each point in the starting phase. Combined with the graph partitioning technique [6], this first stage produces small tight clusters. In the next stage, the algorithm performs the agglomerative process. It tries to merge the small tight clusters

based on the relative inter-connectivity and relative closeness; both are locally normalized by quantifiers. The approach used by *CHAMELEON* provides the solution for finding clusters with irregular shapes, sizes and densities. It achieves this by using dynamic modeling that requires the sets of data to be constructed as a sparse graph in the early stage. Constructing the sparse graph from the data has proven to be time consuming, and therefore inefficient to be used when processing large sets of data. However, this algorithm can be used as an initial stage in processing data incrementally. In our work, *CHAMELEON* will be used to build the initial clusters. The best representation of the initial clusters is crucial because when processing a new data point, the clustering decision will be based on these clusters. Medoid-based approaches, such as *CLARANS* [7] and *PAM* [8] try to find representative points (medoids) to minimize the sum of the distances of points from the closest medoid. This offers two advantages: (1) it solves the metric spaces-specific data problem faced by the centroid approaches and (2) it eliminates outliers by selecting the medoids based on the location of the predominant fraction of points inside a cluster. However, this method fails when dealing with natural data where points in a given cluster are closer to the center of another cluster than to the center of their own cluster. A clustering approach similar in some aspects to our work is used by *DEMON* [9]. *DEMON* is an algorithm that takes the temporal aspect of data evolution into account and allows an analyst to "mine" relevant subsets of the data. This algorithm has the limitation that it uses the *windowing approach* when processing the new data. Furthermore, *DEMON* assumes a supervised scenario (all data has been previously labeled) and it deals with only a restricted number of points.

3 Algorithm

The algorithm we have developed was initially based on *CHAMELEON*. The reason for selecting *CHAMELEON* was its ability to handle clusters of different shapes and sizes with high accuracy. Unfortunately, the processing done by *CHAMELEON* utilizes a *k-nearest neighbor approach* [10], which has two major drawbacks: it requires a significant amount of processing time and it does not scale well to large datasets. Furthermore, *CHAMELEON* does not incorporate incremental processing, hence the algorithm needs to reprocess old data points when the new data arrives. This makes it unsuitable for dynamic sets where the data is observed as a sequential stream of information. However, for our clustering algorithm to be effective, it is desirable to generate an initial set of clusters which closely mirror the groupings found in the data. Hence, our algorithm generates the initial set of clusters using the *CHAMELEON* approach. Once the initial set of clusters is developed, the processing involved in our approach is completely different from that of *CHAMELEON*.

3.1 Cluster Representation

We developed a data structure specifically designed to allow fast, incremental updates. The data structure has two levels. The first level stores all the data points observed while the second level stores only the summary definitions of the clusters. To reduce the computational complexity of the method most of the processing is done at the second level

of the data structure which is stored in the memory. Fundamental to our work is the cluster representation. Each cluster representation contains three dynamic sets of points: an exemplar set, a predictor set and a knowledge repository set. The exemplar points are used to define the basic summary representation of the cluster and each exemplar point has three attributes: reinforcement, age and coverage. All three attributes are dynamic and indicate the “contribution” of the exemplar to the cluster definition. The reinforcement is computed based on the number of new data points observed which have been similar to the exemplar. The age of the exemplar is defined by the time interval over which the exemplar has been consistently reinforced. The coverage is determined by an analysis of the distance between the exemplar and all other exemplars used to define the cluster. Apart from the three attributes, each exemplar has a link to a list of points which are associated with the exemplar at the first level of the data structure. Thus, if a very detailed analysis of the data previously observed is required, the algorithm only needs to retrieve the list from the disk. The exemplar points are extracted from the initial set of clusters generated by the algorithm. From each cluster discovered in the data, the algorithm extracts k points which best capture the shape and size of the cluster. The points are extracted based on reinforcement and spread (average distance to the other k points). The k points are sorted and the best 10 k points (that maximize both reinforcement and average distance) become the exemplar points representing the cluster. As more data becomes available the exemplar points representing the cluster may change to reflect the new trends in the data. In our work we have investigated a range of values for k before choosing $k=10$. The results from the experiments indicate that while larger values such as $k=15$ do not impact significantly on the algorithm processing, the accuracy gained from the extra points was not statistically significant. On the other hand, a smaller k value such as $k=5$ was insufficient to represent some of the more unusual types of clusters that were used in the experiments. The novel aspect of the cluster representation is the use of *predictor* points and a knowledge repository. These two sets of points are used for three reasons: (1) to augment the cluster representation, (2) to adapt the cluster representation to any changes detected in the data and (3) to maintain a history of the information observed. The *predictor* points are used to augment the exemplar points that are used in the summary definition of the cluster. Their purpose is to allow the tracking of changes and prediction of future data. Initially, this set of points are the next best k -points aside from the *exemplar* points to represent the cluster. The selection of the *predictor* points is different from the *exemplar* selection. The requirements for the *predictor* points are that (1) the points need to capture the temporal aspect of the cluster representation and (2) the points should be located some distance between the existing exemplar and predictor points to ensure a good spread of the data representation. Each new data point observed is a potential predictor point candidate and the algorithm has an in-built preference for selecting points which are positioned near the border of the cluster. This was done specifically to allow the close tracking of any changes found in the data. The drawback of this approach is that in cases where the data is noisy the predictors are frequently changed before a reliable subset is found. In general, however, the approach is effective as the clusters were accurately updated to mirror the changes in the data. The predictor selection is done as follows. Each time the algorithm reads a new data point, it calculates the distance between that point and the

exemplar points and stores each of the calculations into the EX-DISTANCE-LIST. It also performs a similar calculation with the predictor points and stores the calculation into the PRED-DISTANCE-LIST. From both the EX-DISTANCE-LIST and PRED-DISTANCE-LIST, the program retrieves the minimum value, which indicates the minimum distance between the new point to the exemplar points and the predictor points. If the minimum distance is less than the average distance of the cluster, the new data point should be considered as a predictor point. Using this approach conforms to the two requirements above. First, because the candidate point is chosen from the new data point, it is able to capture the temporal information of the cluster. Second, performing the distance checking between the predictor and the exemplar points guarantees the wide spread of the new point. The *knowledge repository* has the purpose of storing old data knowledge in the form of “old predictor and old exemplar points”. These points can be used to effectively deal with a recurrent trend in the data. It also reduces the complexity of the processing because we do not need to rediscover clusters and cluster information which existed in the past. Lastly, by keeping the old cluster representation, we are able to produce a continuous interpretation of the cluster definition from time to time.

3.2 Incremental Processing

Most of the processing takes place at the second level of the structure and the processing is of an incremental nature (point by point). The assumptions made were as follows: (1) an initial training set is available at the start of processing and (2) the data observed after the training stage takes the form of a sequential stream of points. The approach we use is unsupervised and all the data processed is assumed to be unlabeled. In the training stage the system generates the set of initial clusters using the CHAMELEON approach. The clusters are processed and the exemplar and predictor points are extracted for each cluster. After the training stage is completed, the algorithm uses incremental learning to handle the rest of the data observed. The program reads the new data points and tries to assign the points to existing clusters if they satisfy all the necessary requirements. The requirements are based on a comparison of the distance between the new data point to the cluster’s representation points and the average distance between the representation points. When locating the closest cluster, the general approach is to assign new data points to a cluster according to the closest k nearest exemplar. An average distance between the k nearest exemplar and the new point is derived for each set of clusters and, the cluster with the smallest average distance between the exemplar and the point is the cluster the new point is most likely to be assigned to. If the distance between the new point and the exemplar or the centroid is greater than the average distance between points in a cluster, the algorithm assigns the new point as a new cluster. Next, it will update the reinforcement of the exemplar point and the predictor point lists. The algorithm attempts to match the new point to a predictor or an exemplar of an existing cluster. The point can either reinforce an existing exemplar or predictor point or form the basis of a new cluster. As the data is observed, we use the set of predictors and exemplar points to update the cluster definition. Over time, the points in the predictor list may become more significant than the exemplar points. The algorithm updates the representation points by considering the predictor points as candidate points. First, it inserts the points from the predictor list that meet *reinforcement threshold* into the exemplar point list.

Next, it retires the exemplar points that are no longer important into the repository list. The retirement of an exemplar point is based on both the *reinforcement* and the *age* attributes of the exemplar.

The predictor point update is similar to the exemplar point processing. A predictor point's reinforcement value is updated when that point is located closer to the new point when compared with the closest exemplar. The algorithm searches for a point which has the closest distance to the new point. If such a point exists, then the algorithm updates its reinforcement value. Updating the reinforcement information of the predictor point is essential because it indicates the importance of that point in a particular time frame. If the level of reinforcement of a predictor point is low then the point is "retired" in the knowledge repository. On the other hand if the level of reinforcement is high, a predictor can be promoted to an exemplar point. The reinforcement of points is driven by the underlying trend in the data. In some cases, if the trend in the data is recurrent, old points from the knowledge repository may be reused in cluster definitions. The algorithm always checks the repository whenever a new predictor point is added to the predictor list. If the new candidate is similar to a past predictor/exemplar, then the old point is reused instead of the new candidate. After the reinforcement update, the program checks for special cases of the current cluster representation. Hence the algorithm checks whether it should merge two clusters or split the current cluster. If the new data point is not assigned to any clusters, then a new cluster is created.

4 Results

The algorithm was tested using several generated datasets. The datasets were created using VidGen, a tool that allows the user to specify the shape, size and density of a cluster as well as the number of clusters to be generated for the dataset. Furthermore, the data had a time index which allowed the simulation of (1) different trends over time and (2) different reinforcement patterns. In this section we present detailed results from 2 scenarios. Each scenario consisted of a training set (from which the initial set of clusters is generated) and test set (where the data arrives in the form of a sequential stream) ranging from 100,000 to 2,000,000 points. The test set was further divided into several consecutive phases to allow for a more detailed analysis of the algorithm performance (hence we present the results for each individual phase in the test set). The aim of each scenario was to test the different requirements of the algorithm: accuracy, data prediction, and computational efficiency. All the results presented have been obtained after 10 runs and an average over the 10 runs is given as the final result. The PC used for the experiments was a P4-1.8Mhz with 256Mb of RAM running Windows2000.

4.1 Scenario 1

The scenario was designed to test the speed and clustering accuracy of the program when dealing with large datasets, specifically 2,000,000 points. The test was carried out on 3,000 training data points and 2,000,000 incoming data points depicted in Figure 1. The incoming data arrived in three phases. The accuracy of the clustering algorithm and the time of processing will be shown in tables for each phase. The table explanation is as

Table 1. Test With Large Data Set - First Phase

	Training Set		Combined Sets Phase One		
	Original	New Approach	Original	Added Point	New Point
Cluster 1	1135	1135	1135	-	1135
Cluster 2	1095	1095	1095	-	1095
Cluster 3	822	822	822	-	822
Cluster 4	-	-	-	691197	689062
Dropped	-	-	-	-	-
Time	-	-	-	-	5 mins 13 secs

Table 2. Test With Large Data Set - Second Phase

	Combined Sets Phase Two		
	Original	Added Point	New Point
Cluster 1	1135	-	1135
Cluster 2	1095	681087	682182
Cluster 3	882	-	882
Cluster 4	691197	-	689062
Dropped	-	-	-
Time	-	-	11 mins 28 secs

Table 3. Test With Large Data Set-Third Phase

	Combined Sets Phase Three		
	Original	Added Point	New Point
Cluster 1	1135	-	1135
Cluster 2	682182	-	682182
Cluster 3	882	667647	668529
Cluster 4	691197	-	689062
Dropped	-	-	-
Time	-	-	16 mins 26 secs

follows. The *training set* column contains the information about the training stage and is divided into two sub-columns. The *original* sub-column indicates the actual number of data points of each cluster while the *new approach* sub-column indicates the number of data points that is clustered correctly to the cluster. The *combined sets phase one* column contains the result produced in phase one and is itself divided into three sub-columns. The *original* sub-column indicates the number of data points of each cluster before the new data arrives. The *added point* sub-column indicates the number of data points added to a particular cluster. Lastly, the *new approach* sub-column contains the number of points of each cluster after the new data points are added. All other tables follow the same format described above.

Phase One: In the first phase, the incoming data arrives in the area that contains no existing clusters (see Figure 2). The algorithm creates a new cluster and classifies all the incoming data into a single cluster. The processing time to classify the first phase is five minutes and thirteen seconds.

Phase Two: In the second phase, the data arrives in an area near cluster two. Figure 3 illustrates the current exemplar points when all the new data points have been classified. The total time required to process the first and second phase is eleven minutes and twenty eight seconds.

Phase Three: In the final phase, the data arrives in an area near cluster three. Figure 4 illustrates the current exemplars of cluster three after the cluster representation is updated. The overall time to classify the three phases is sixteen minutes and twenty six seconds.

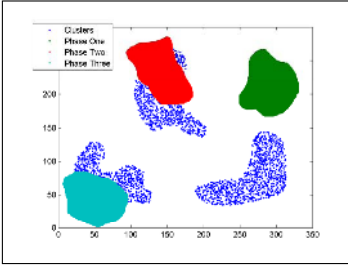


Fig. 1. Combined Phases - Training & Test Data

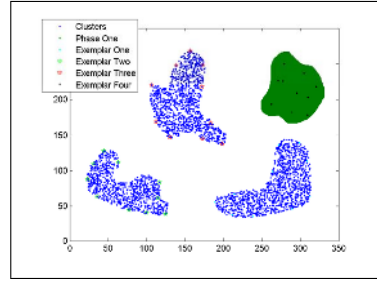


Fig. 2. Phase One - Training & Test Data

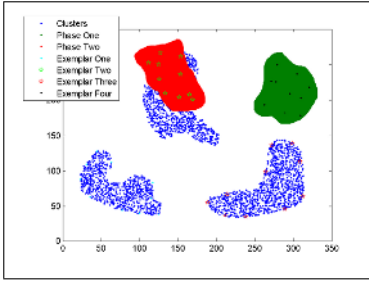


Fig. 3. Phase Two - Training & Test Data

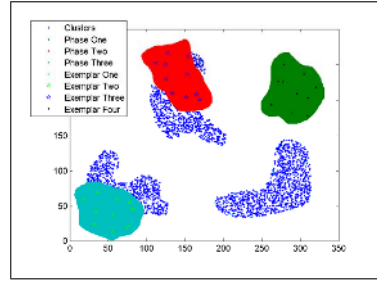


Fig. 4. Phase Three - Training & Test Data

5 Scenario 2 – Comparison with CURE

Scenario two was designed to compare our approach with an existing algorithm. We selected CURE for two reasons. It is an algorithm that generally has good performance in terms of handling data with clusters of different shapes and sizes, as well as in terms of speed. We also intended to compare our approach with CHAMELEON but the memory and computational requirements of CHAMELEON made testing with large datasets unfeasible. CURE does the clustering process in one phase while our approach processes data incrementally. As a result, the test cases are designed to fulfill the requirement of the two algorithms. We did the testing using two sets of data, starting from 20,000 points to 100,000 points. Figure 5 illustrates the training data and the incremental data used in testing. The shape of the clusters throughout the testing is the same. The input data for CURE is the combination of the training data and the incremental data. For our approach, we start by processing the training data and then incrementally process the rest of the test data. Figure 6 illustrates the exemplar points extracted from the clusters.

Testing Using 20,000 Points. Figure 7 illustrates the result produced by CURE. When compared with our approach, CURE fails to adjust the cluster definition to accurately reflect the trend shown by the new data points and this reduces the CURE overall accuracy (most of the new points are misclassified). This result is not surprising because the size of the incremental data is larger than the training, and CURE prefers to select clusters that have high density. On other hand, our algorithm classifies the data points

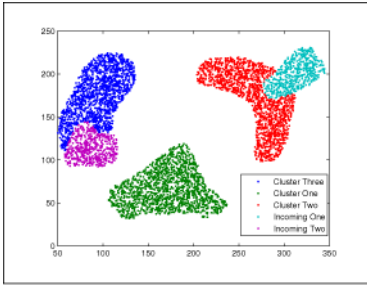


Fig. 5. Scenario 2 - Test & Training Set Combined

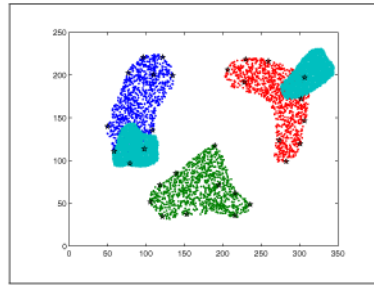


Fig. 6. Scenario 2 - Exemplar Points Extracted by Our Approach

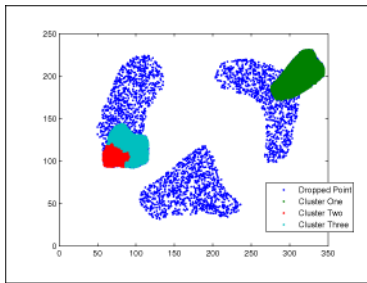


Fig. 7. CURE - 20,000 Points Test

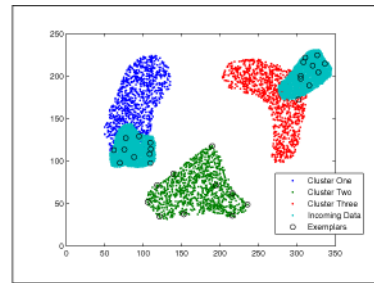


Fig. 8. Our Approach - 20,000 Points Test

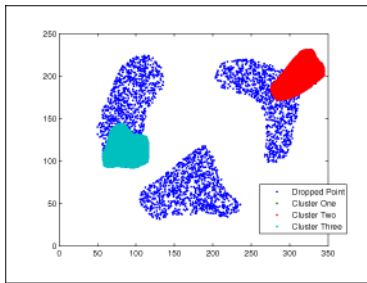


Fig. 9. CURE - 100,000 Points Test

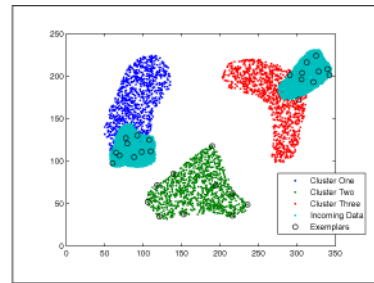


Fig. 10. Our Approach - 100,000 Points Test

with high accuracy. Our algorithm does not rely heavily on the data density. It considers density as one of the trends of the data, so data can have many levels of density in different time frames. Figure 8 illustrates the result produced using our approach. Table 4 summarizes the overall result. Finally, our method is faster when compared with CURE.

Testing Using 100000 Points. The second test was carried out using 100,000 points. The same approach was applied where the size of the testing data remains the same

Table 4. Scenario 2 Result - 20000 Test

Combined Sets 20000 Test			
	Original	Cure	Our Approach
Cluster 1	11019	2728	10895
Cluster 2	11026	5765	10934
Cluster 3	1071	0	1071
Time	-	7 mins 23 secs	47 secs
Dropped	-	14623	-

Table 5. Scenario 2 Result - 100000 Test

Combined Sets 100000 Test			
	Original	Cure	Our Approach
Cluster 1	50784	37913	49784
Cluster 2	51774	38366	50374
Cluster 3	1071	0	1071
Time	-	2 hrs 10 mis	1 min 3 secs
Dropped	-	25208	-

while we increase the size of the incremental data. The result is similar to the previous test (see Figures 9 and 10) . However, there is one more major limitation that is not addressed by CURE. In Table 5, we can see that it takes two hours and thirteen minutes to process 100,000 data points, which makes CURE unsuitable to handle large data sets. Our approach deals only with the exemplar and predictor points. As a result it is computationally less intensive, because the processing is done incrementally rather than simultaneously.

6 Conclusions

In this paper we have presented an algorithm to cluster large dynamic datasets. The research has two novel aspects: the use of a new cluster representation which combines both exemplar and predictor points and the integration of drift tracking techniques in the incremental processing of the data. The algorithm has been tested with several synthetic datasets and its performance was compared with the CURE algorithm. The results show the algorithm has good accuracy and it is both faster and more accurate when compared with CURE. Future work will be done on improving the overall speed of the algorithm and on using more advanced tracking techniques to allow for a better adaptation of the clusters to changes in the data.

References

1. Bradley, P.S., Fayyad, U.M., Mangasarian, O.L.: Data mining: Overview and optimization opportunities. Technical report, Microsoft Research Lab (1998)
2. Berkhin, P.: Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA (2002)
3. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: A new data clustering algorithm and its applications. *Data Min. Knowl. Discov.* **1** (1997) 141–182
4. Guha, S., Rastogi, R., Shim, K.: CURE: an efficient clustering algorithm for large databases. In: *Proceeding of ACM SIGMOD International Conference on Management of Data*, Seattle, WA, USA (1998) 73–84
5. Karypis, G., Han, E.H.S., Kumar, V.: Chameleon: Hierarchical clustering using dynamic modeling. *Computer* **32** (1999) 68–75
6. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.* **20** (1998) 359–392
7. Ng, R.T., Han, J.: Clarans: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering* **14** (2002) 1003–1016

8. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, New York (1990)
9. Ganti, V., Gehrke, J., Ramakrishnan, R.: Demon: Mining and monitoring evolving data. *IEEE Transactions on Knowledge and Data Engineering* **13** (2001) 50–63
10. Therrien, C.W.: Decision estimation and classification: an introduction to pattern recognition and related topics. John Wiley & Sons, Inc. (1989)

Birds of a Feather Surf Together: Using Clustering Methods to Improve Navigation Prediction from Internet Log Files

Martin Halvey, Mark T. Keane, and Barry Smyth

Adaptive Information Cluster, Smart Media Institute,
Department of Computer Science, University College Dublin, Belfield,
Dublin 4, Ireland

{martin.halvey, mark.keane, barry.smyth}@ucd.ie

Abstract. Many systems attempt to forecast user navigation in the Internet through the use of past behavior, preferences and environmental factors. Most of these models overlook the possibility that users may have many diverse sets of preferences. For example, the same person may search for information in different ways at night (when they are pursuing their hobbies and interests) as opposed to during the day (when they are at work). Thus, most users may well have different sets of preferences at different times of the day and behave differently in accordance with those preferences. In this paper, we present clustering methods for creating time dependent models to predict user navigation patterns; these methods allow us to segment log files into appropriate groups of navigation behaviour. The benefits of these methods over more established methods are highlighted. An empirical analysis is carried out on a sample of usage logs for Wireless Application Protocol (WAP) browsing as empirical support for the technique.

1 Introduction

One of the key challenges in adaptive hypermedia and personalization is to properly capture user preferences based on their past behavior, explicit and implicit preferences and other environmental factors. When these and other factors are known it becomes more possible to predict what a user is looking for, and for a system to automatically adapt to the user using such predictions. In this paper, we examine navigation in the context of the mobile-Internet with a view to predicting user preferences for certain sites based on environmental factors (i.e. time of the week and paths followed). Cotter and Smyth [6] have proposed that users should not have just one set of preferences but rather groups of preferences, characterizing their browsing *personality* in different contexts. The fundamental insight behind the present work is that a user navigating during office hours would have different preferences than that same user on the weekend. As such, the research challenge is to determine the boundaries between people's distinct personas and to be able to use these categories to predict

preferences. In this paper, we outline methods for clustering together user sessions based on the paths followed in those sessions, where a path is a series of URLs selected within a session. We advance these methods as a means for automatically determining different sets of user preferences, thus providing an important component for any system that wishes to be truly adaptive.

2 Task Context: Navigation on the Mobile-Internet

WAP is the basis for the wireless Internet. It is an open global specification that provides Internet browsing functionality for small hand-held devices such as mobile phones to easily access and interact with information and services instantly. WAP users face additional problems that PC users do not, the screen real estate on a mobile phone is several orders of magnitude smaller than that of PCs. The mobile phones' capabilities are much more diverse than the more standardized PCs (e.g., in display resolution, color capability, operating system features, browser functionality). Mobile phones also have very limited input capabilities, featuring numeric keypads with minimal text entry, unlike the mouse and keyboard options available to PC users. Finally, the content-base of WAP is considerably less diverse and when this content is accessed, users face slow download times and incremental billing costs [16]. While some of the problems, for example slow download times and phone capabilities have been addressed, there is still no definitive solution to aid user surfing via WAP. Here we provide one possible solution.

Intuitively, users' navigation on the weekend when they have leisure days should differ from their navigation on weekdays when they are at work. Following this intuition, our hypothesis is that users surf differently during different time periods, and that these differences can be used to make predictions about user navigation. The main idea behind using clustering, to attempt to determine this split, is to group together sets of paths that users have followed that contain similar types of pages (for example user sessions involving sports sessions should be grouped together, in the hope that these types of pages will display some distinct patterns with relation to time for example and allow the segmentation of the log files to improve navigation prediction). Halvey et al [9] have previously shown that predictive models that are time dependent can greatly improve accuracy in navigation prediction. However the segmentation step in their time-based models were handcrafted rather than automated. Here we propose a more formal method of grouping together user sessions and segmenting log files. The purpose of this work is to aid users' navigation on the Internet. By predicting user navigation with a high degree of accuracy, you can for example aggressively promote a URL so that the URL that a user would like to select is always at the top of a series of menu selections or the desired URL is highlighted in order to help a user find it. To test this hypothesis we analyzed a data set from a mobile Internet portal. By exploiting the uneven distributions in WAP surfing pattern we endeavored to determine whether distinct navigation patterns arose and, if found, whether these patterns could be accurately predicted using Markov models.

3 Background Research

3.1 Clustering

Clustering algorithms have been used in a broad range of applications. In particular we are interested in hierarchical agglomerative clustering algorithms as they are the type of clustering algorithm that we are using in this work. In the area of image segmentation an agglomerative clustering algorithm was applied in Silverman and Cooper [17] to the problem of unsupervised learning of clusters of coefficient vectors for two image models that correspond to image segments. Jain and Dubes have applied the complete-link hierarchical clustering scheme to the problem of object and character recognition [11]. They have also applied the complete link agglomerative clustering algorithm in the area of document retrieval to create a proximity dendrogram for a collection of books [11]. Etzioni has applied hierarchical clustering methods in the area of data mining [7]. For a more complete review of clustering methods and their applications please refer to Jain et al [12].

3.2 Time-Based Analysis

In recent times, there has been an increasing interest in the use of time in conjunction with predictive models. Analyses of time-based patterns of work in office environments are an example. Begole et al [2] attempt to detect and model rhythms of work patterns in an office. Horvitz et al [10] use Bayesian networks built over log data to model time-based regularities in work patterns in order to predict meeting attendance and interruptability.

Time-based analyses of web searching have also been carried out. With the aim of supporting users, Lau and Horvitz [13] have constructed probabilistic models centering on temporal patterns of query refinement to predict how a user would continue their search. Beitzel et al [3] have analyzed search engine queries with respect to time and found that some topical categories vary substantially more in popularity throughout the day; they also found that query sets for different categories have differing similarity over time.

Halvey et al [9] have also recently conducted an analysis of mobile Internet browsing patterns with respect to time. They discovered that users browsing patterns had temporal variations and exploited these patterns to improve navigation prediction accuracy. However much of this work was done by hand and required many steps.

3.3 Web Navigation

Lieberman [14], Cooley et al [5] and Spilioulou [19] have all also presented solutions that take advantage of earlier user experiences to create adaptive Internet websites. Of particular interest to this current work is that other researchers have used Markov models to create predictive models of web navigation. Pirolli [15] has shown that k-means Markov models can be used to forecast user navigation patterns. Zhu et al [21] have used Markov chains, based on past navigation patterns, to predict web page accesses. As was discussed in the previous section navigation prediction in the mobile Internet presents additional problems, Billsus et al [4], Anderson et al [1] and Smyth and Cotter [18] all offer solutions to the problem of navigation in the mobile Internet.

4 Clustering Sessions Based on Paths

The aim of this work is to segment web logs in such a way that making predictions about user navigation becomes simpler and more accurate. The WAP portal that is used can be represented as a tree. It is hoped that the clustering will result in paths from similar sections of the tree being grouped together to reveal information about users who have an interest in that section of the portal. Also paths of similar length may be grouped together, and may also reveal that users who favour longer sessions may have distinct features from users who favour shorter sessions.

4.1 Distance Metrics

To begin with each URL that was selected by a user was represented by a symbol, and a path was then represented by a sequence of these symbols. The first task was to calculate a distance between these paths that were traversed. The first distance metric that we used was a simple Euclidean distance ($\sqrt{\sum_{i=1}^N (p_i - q_i)^2}$) where if $p_i = q_i$ then a value zero is returned otherwise one is returned. So for example for two strings where the first string is “Mark” and the second string is “Martin” the Euclidean distance is $\sqrt{(0+0+0+1+1+1)} = 1.732051$. The second method that was used was Levenshtein Distance or Edit Distance, which is for two strings, s_1 and s_2 , the minimum number of point mutations required to change s_1 into s_2 , where a point mutation is one of either change a character, delete a character or insert a character. For example “Mark” and “Martin” have a Levenshtein Distance of 3, however “Martin” and “Barry” have a Levenshtein Distance of 4. The third and final method that we use is a derivation of Euclidean Distance that for the purposes of this work has been called *Total Euclidean*. As the WAP site in which the navigation took place is a tree we performed a depth first search on the tree and assigned each node incrementally an integer value. The Euclidean distance equation is then applied. As stated previously, if the two nodes are the same then the distance between them is zero. However if two nodes are different then the difference between them is the difference between the integer values assigned to their nodes. In this way paths in branches of the tree closer together will have a distance that is shorter than those in branches of the tree that are further apart.

4.2 Clustering Paths

To cluster the paths we used hierarchical agglomerative clustering methods, single link, complete link and average link algorithms were implemented. A fixed number of clusters were not set for these methods; instead these algorithms were given a stopping parameter. The parameter chosen was that when d_1 , the average distance between clusters, is less than half of d_2 , the average distance between elements in the clusters, then the algorithm should stop i.e. when $d_1 > (d_2)/2$ stop. Initially the parameter was when $d_1 > d_2$, however due to some outlying nodes the majority of the clustering methods did not halt until all of the elements were members of one large cluster.

4.3 Finding Time Related Segments

As stated previously Halvey et al [9] have also recently conducted an analysis of mobile Internet browsing patterns and discovered that users browsing patterns had temporal variations. Accordingly each of the clusters formed was analysed to see if there is a distinct or dramatic rise or fall in the number of hits that the clusters have received with respect to time (either days or hours). If such an anomaly occurs sessions for that time period are extracted from the log files and a distinct predictive model is created for that time period.

5 Predicting User Navigation Using Log Data

5.1 Predicting User Navigation

To test whether these approaches could be used successfully to automatically segment web logs (and later be used to make predictions about user navigation and ultimately aid that navigation) we analyzed a data set from a mobile Internet portal. This data, gathered over four weeks in September 2002, involved 1,168 users and 147,700 individual user sessions using WAP phones to access a major European mobile Internet portal. Using the distance metrics and clustering algorithms outlined earlier the paths followed by users in the WAP portal were clustered. However not all nine possible combinations of distance metrics and clustering methods formed clusters, the Euclidean and Levenshtein distance methods formed clusters using both the average and complete link algorithms, the *Total Euclidean* distance method formed clusters using the complete link algorithm. The log files were then segmented according to the clusters of paths followed.

To evaluate the success of different clustering methods we constructed Markov models, similar to Zhu [21], for each of the segmented data sets, as well as models for all of the data. Each of these models was then used to predict the next URL clicked in a path given the current path for each of the segmented data subsets and for values of k between 1 and 5. Five was chosen as an upper limit as some of the segments contained only a small number of sessions and also some of the segments contained sessions where only short paths were followed. In each WAP menu the user has approximately seven selections (including going back to the previous page) from which they can choose, therefore the result of random recommendations should be approximately one in six a baseline 0.167%. However for these experiments a fully connected graph was assumed, to take into account instances where users used a bookmark or typed in a URL in the mid-session for example. As this theoretically gives users a one in 256 choice, there is a baseline of approximately 0.004% accuracy. The models created were then tested on a sample of random sessions from the log files to calculate the accuracy of the models, for these experiments we consider accuracy to be the proportion of URL's that are correctly predicted by the Markov models. The results of these experiments are shown in figure 1. The accuracy for the models created using the data from the segmented log files is contrasted with the accuracy for the predictive model built using all of the log file data. Overall, one major conclusion can be drawn

from these models about our ability to predict navigation behavior and the success of our model for segmenting the log files. That is, if one tries to predict navigation during a particular set of sessions using the full data set the predictive accuracy of the model is not as accurate as the model that corresponds to that set of sessions.

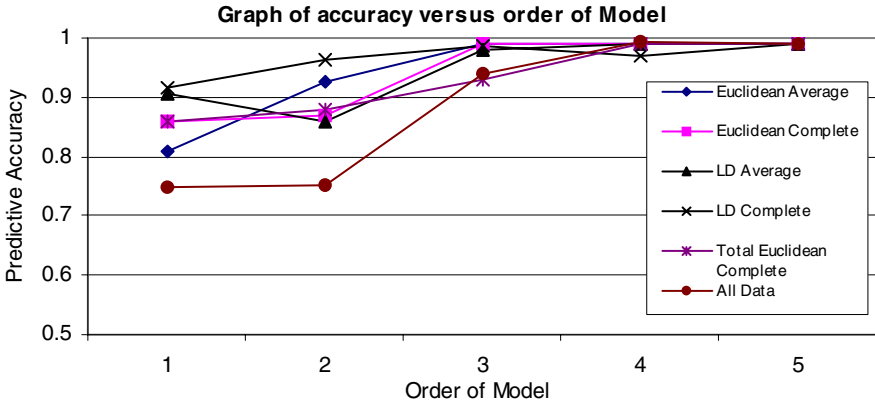


Fig. 1. Graph illustrating the predictive accuracy for each of the segmented data sets with respect to k

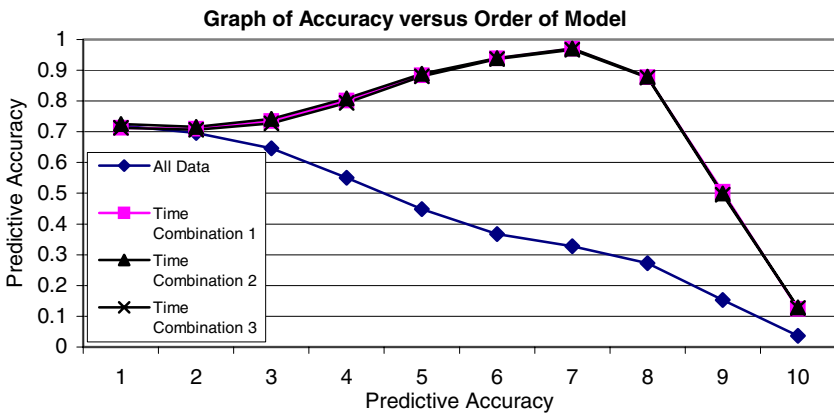


Fig. 2. Graph illustrating the predictive accuracy for each of the segmented data sets with respect to k

As outlined in section 4.2 the clusters formed were then analyzed to see if there were any time dependencies in the clusters. Three of the five sets of clusters had the same time dependencies, the other two clusters found slight variations of the first time

relationship. Once again the log files were segmented, however on this occasion the segmentations were based on the time dependencies discovered. Also as was done previously Markov Models were formed, however for these segments the maximum value of k was 10 as there were no significantly small segments and all of the segments contained sessions with various path lengths. As before the models created were then tested on a sample of random sessions from the log files to calculate the accuracy of the models, and once again for these experiments we consider accuracy to be the proportion of URL's that are correctly predicted by the Markov models. The results of these experiments are shown in figure 2. The same conclusions can be drawn from the results in figure 2 as were concluded in figure 1. However, it may be noted that after a certain order of model the accuracy begins to tail off. However, this is not really a concern as most WAP users favour shorter sessions over longer sessions according to the Universal Law of Web Surfing [8].

5.2 Empirical Evaluation of Predictive Models

We can also put the knowledge we have gained from our Markov modeling to work in assessing how it would impact personalization of a mobile portal. Smyth & Cotter [18] have developed the Click-Distance Model to estimate the likely navigation gains in a given mobile-portal when personalization is used for menu re-ordering. Figure 3 illustrates the results of the click distance analysis for the cluster based models and Figure 4 illustrates the results of the click distance analysis for the time-based models.

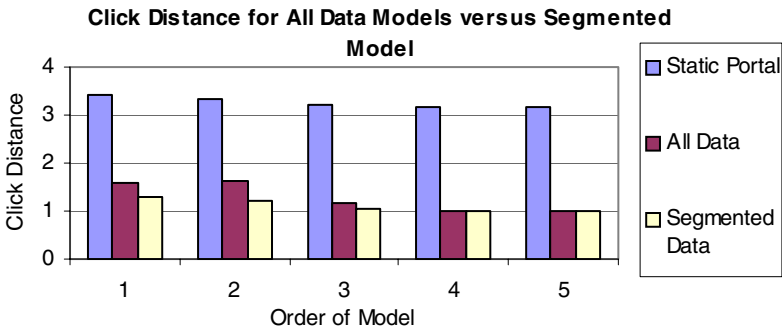


Fig. 3. Results of click distance analysis for static portal, Markov models using all data and Markov models created using clustered data

These results can be summarised in a few points. The use of any navigation data reduces mean click-distance significantly in comparison with when it is not used; therefore personalisation using navigation patterns helps. Also the effectiveness of these models improves with the order of the Markov models. Finally, in nearly all of the models, the models based on the clustered data results resulted in shorter click distances than the models based on the whole data set.

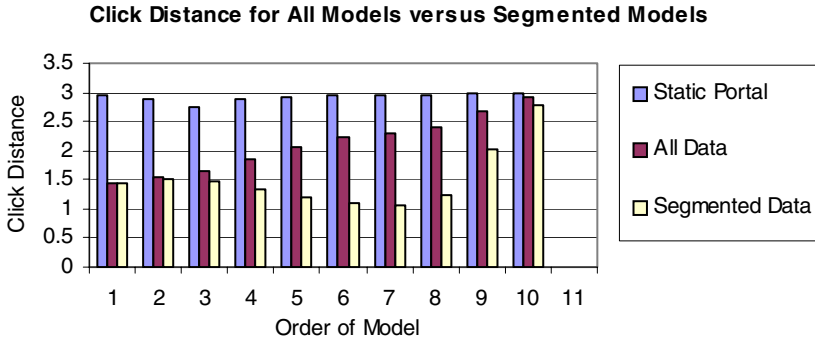


Fig. 4. Results of click distance analysis for static portal, Markov models using all data and Markov models created using data segmented based on time

6 Conclusions and Future Work

Users have different needs and goals at different times and as such user navigation patterns in the Internet are time dependent. In this paper we have presented a method that takes advantage of this phenomenon by automatically segmenting log file data, based different time periods and different goals. This has been confirmed by an analysis of a WAP usage log file. Clusters of usage patterns were created, and from these clusters Markov models were learnt. These predictive models were compared with Markov models built over all of the log data. The predictive accuracy for the Markov models for the explicit time periods and clusters was far greater than the accuracy of the other models constructed. These more accurate models can also be used reorganize the structure of the portal to reduce the click distance and thus reduce the amount of effort for users. These results support our hypothesis as well as highlighting the potential of such data segmentation for aiding user navigation creating truly adaptive systems. Consequently, there is a huge potential benefit to Internet users for usage of such techniques, in particular mobile Internet and WAP users who encounter additional problems that desktop users do not encounter [16], which we have highlighted previously.

Additionally predicting user navigation could be used to improve download times. While a server is not busy, predicted pages could be pre-fetched for users to reduce download times, this would be particularly useful for mobile users for whom download times are a particular problem [16]. Also predicting and pre-fetching pages could also reduce the load on servers. As this is an initial attempt at segmenting log data according to time there are, of course, other extensions that can be made to this work. Firstly this segmentation of the data could quite easily be used in conjunction with some other predictive model, for example the ClixSmart navigator [18] to make more accurate predictions about user navigation and adapting portal structure to the needs of users. With the integration of some of these techniques we may be able to discover other temporal segmentations and make even more accurate recommendations.

In this paper we have outlined new methods to aid users of both the mobile-Internet and Internet. This study is a new direction in Internet navigation prediction and will hopefully lead the way in finding the solution to what is a very difficult problem.

Acknowledgements

This material is based on works supported by the Science Foundation Ireland under Grant No. 03/IN.3/I361 to the second and third authors.

References

1. Anderson, C.R., Domingos, P. & Weld, D.S., Adaptive Web Navigation for Wireless Devices, Proceedings of Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01), 2001.
2. Begole J., Tang J.C. & Hill B., "Rhythm Modeling, Visualizations and Applications", Proceedings of the 2003 Symposium on User Interface Software and Technology (UIST 2003), pp. 11-20, 2003.
3. Beitzel S., Jensen E., Chowdhury A., Grossman D. & Frieder O. HoURLy analysis of a very large topically categorized web query log. Proceedings of the 27th annual international conference on Research and development in information retrieval, pp 321 – 328, 2004.
4. Billsus D., Brunk C., Evans C., Gladish B. & Pazzani M. Adaptive Interfaces for Ubiquitous Web Access, Communications of the ACM, Vol 45, No 5, 2002.
5. Cooley R., Mobasher B. & Srivastava J. Web Mining : Information and Pattern Discovery on the World Wide Web, Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), Newport Beach, CA, November 1997.
6. Cotter P., & Smyth B. "PTV: Intelligent Personalised TV Guides". Proceedings of the 12th Innovative Applications of Artificial Intelligence (IAAI-2000) Conference. AAAI Press, 2000.
7. Etzioni, O. The World-Wide Web: quagmire or gold mine? Communications of the ACM 39, 11, 65–68
8. Halvey M, Keane M.T. & Smyth B. "Mobile Web Surfing is the same as Web Surfing", Communications of the ACM, 2005. (Accepted; In Press).
9. Halvey M., Keane M.T., & Smyth B., "Predicting Navigation Patterns on the Mobile-Internet using Time of the Week", World Wide Web 2005, (Accepted; In Press)
10. Horvitz E., Koch P., Kadie C.M., & Jacobs A. Coordinate: Probabilistic Forecasting of Presence and Availability In: Proceedings of the Eighteenth Conference on Uncertainty and Artificial Intelligence, Edmonton, Alberta. Morgan Kaufmann Publishers, pp. 224-233, 2002.
11. Jain A. K., & Dubes R. C. Algorithms for Clustering Data. Prentice-Hall advanced reference series. Prentice-Hall, Inc., Upper Saddle River, NJ, 1988.
12. Jain A., Murty M.N., & Flynn P. Data clustering: A review. ACM Computing Surveys, 31(3):264–323, September 1999.
13. Lau T. & Horvitz E. Patterns of Search: Analyzing and Modeling Web Query Refinement. Proceedings of the Seventh International Conference on User Modeling, 1999.

14. Lieberman H. Letizia: An Agent That Assists Web Browsing, International Joint Conference on Artificial Intelligence, Montreal, August 1995.
15. Pirolli P. "Distributions of Surfers' Paths through the World Wide Web: Empirical Characterizations." *The Web Journal*, 2 : 29-45, 1998.
16. Ramsay M., Nielsen J. Nielsen Report, "WAP Usability Deja Vu: 1994 All Over Again", 2000.
17. Silverman J. F., & Cooper D. B. Bayesian clustering for unsupervised estimation of surface and texture models. *IEEE Trans. Pattern Anal. Mach. Intell.* 10, 482-495, 1998.
18. Smyth B. & Cotter P. "The Plight of the Navigator: Solving the Navigation Problem for Wireless Portals". Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web Systems. Malaga, Spain, pp 328-337, 2002.
19. Spiliopoulou M. The laborious way from data mining to Web log mining. *International Journal of Computer Systems Science and Engineering*, 14(2):113-125, 1999.
20. Wu Z. & Leahy R. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 15, 1101-1113, 1993.
21. Zhu J., Hong J., & Hughes, J.G. Using Markov models for web site link prediction. *ACM Conference on Hypertext/Hypermedia*, 2002.

Alarm Clustering for Intrusion Detection Systems in Computer Networks

Giorgio Giacinto, Roberto Perdisci, and Fabio Roli

Department of Electrical and Electronic Engineering, University of Cagliari,
Piazza D Armi - 09123 Cagliari, Italy
{giacinto, roberto.perdisci, roli}@diee.unica.it

Abstract. Until recently, network administrators manually arranged alarms produced by Intrusion Detection Systems (IDSs) to attain a high-level description of threats. As the number of alarms is increasingly growing, automatic tools for alarm clustering have been proposed to provide such a high level description of the attack scenario. In addition, it has been shown that effective threat analysis require the *fusion* of different sources of information, such as different IDSs, firewall logs, etc. In this paper, we propose a new strategy to perform alarm clustering which produces unified descriptions of attacks from multiple alarms. Tests have been performed on a live network where commercial and open-source IDSs analyzed network traffic.

Keywords: Computer Security, Intrusion detection, Clustering.

1 Introduction

At present, a number of commercial, open-source, and research Intrusion Detection Systems (IDSs) tools are available. They differ in the way intrusions are detected, and in the available options allowing further alarm processing. Among them, network misuse detectors are widely used in many organizations for their ability in detecting well-known patterns of intrusions.

Network misuse detectors analyse network traffic looking for packets whose characteristics match the "signature" of known attacks. As soon as a signature is matched, an alarm is raised. As signature matching is performed on a packet basis, alarms provide a powerful source of fine-grain information related to suspect activities in the protected network. In order to gain an understanding of the intrusions against the protected network, a network administrator needs to arrange these alarms to produce a high-level description of the threat. As the number of alarms is increasingly growing, it is not feasible for a network administrator to manually arrange the huge volume of alarms. Recently a number of alarm clustering products have been proposed to provide such a high level description of the attack scenario [1]. The aim is to manage the large number of so-called *elementary* alarms produced by IDSs, by their fusion in higher-level alarm messages. The source of such a large number of alarms is motivated by

the nature of some categories of attacks which send a large number of malicious packets. As signature-based IDSs produce an alarm for each malicious packet, alarm flooding may occur. Alarm clustering can also be used to *fuse* alarms from different sensors. The use of multiple *complementary* Intrusion Detection technologies can provide the following benefits: i) for a given attack, different IDSs may produce different outputs; ii) for a given attack, only a limited number of IDSs might be able to detect it; iii) the fusion of alarms raised by different IDSs may attain more comprehensive information about intrusion attempts than that attained using a single IDS technique. Therefore, the proposed multiple-sensor environment is made up of a number of IDSs (e.g., commercial and open-source products), and the measurements to be fused are the elementary alarms raised by each IDS. This paper proposes an *on-line* alarm clustering algorithm whose output is a set of *meta-alarms*. During the operation of the IDSs, the alarms are analysed and clustered. When no further alarm can be clustered to an existing group, the related *meta-alarm* is output to the administrator.

Meta-alarms provide a network administrator with summary information about the attack and the related alarm messages produced by IDSs. This information can be further used by higher-level modules that perform *multiple-step attack* scenario reconstruction and threat analysis.

At present, a few works on alarm clustering and correlation have been presented [2-5]. With respect to the related work, in the present paper a novel *on-line* alarm-clustering algorithm is proposed. The objective is to achieve alarm volume reduction by fusing alarms produced by different sensors in consequence of a given attack. In particular, the main contribution is the introduction of a learning phase, which aims at extracting the attack class(es) an attack description belongs to. This attack description classification process allows to cluster alarms seemingly produced by different attacks but belonging to the same alarm thread.

The paper is organized as follows. Section 2 presents the details of the proposed alarm clustering algorithm. Some results attained on a test network with commercial and open-source IDSs are reported in Section 3. In particular, the structure of the meta-alarm is presented which can summarize a large number of elementary alarms. Conclusions are drawn in Section 4.

2 The Proposed Alarm Clustering Algorithm

In this section, we present our alarm clustering algorithm designed to process the sequence of alarms produced by IDSs, and then produce *meta-alarms*, i.e. summary descriptions of events obtained by aggregating correlated alarms produced by various IDS sensors. Such a summary information can be provided by the attack class the alarms refer to. The alarm class provides an effective high-level information [6] that can be used by higher-level modules that perform multiple-step attack scenario reconstruction. As an example, let us consider the three attack classes used in our experiments, i.e., *portscan*, *web-scan*, and *DoS* (Denial of Service). A *portscan* attack is performed by sending a very large

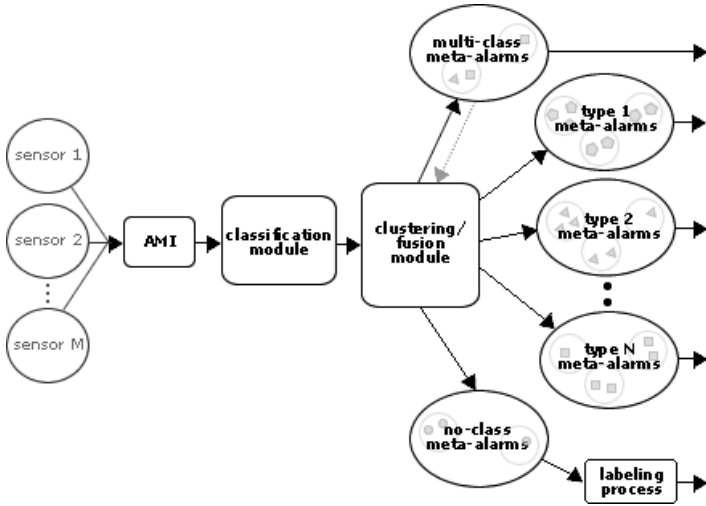


Fig. 1. Alarm Clustering Module

number of TCP or UDP packets to different ports in order to spot whether a service is bound to a certain port or not. In a similar way a *webscan* attack is performed by sending a sequence of HTTP queries to a web server (the victim) looking for vulnerable web applications. DoS attacks, instead, are usually performed by sending a large number of properly crafted packets to a victim host trying to exploit vulnerabilities which can cause the victim host or application to crash. Each meta-alarm is further described by a number of features needed to uniquely identify the event, such as start and stop times, source and destination IP, etc. In addition, the identifiers of the aggregated alarm logs are reported for further inspection. The proposed system results particularly suitable in aggregating alarms produced by those kinds of attacks which cause the IDSs to produce a high number of alarms. In the following, we will refer primarily to signature-based Network-IDS (NIDS) sensors, as it is the most widely used type of IDS sensors. Nevertheless, the reported discussion can be extended to other ID techniques. We will provide an overview of the architecture of the proposed alarm-clustering module first, then going into the details of each of the components. Figure 1 depicts a schema of our alarm clustering system.

The first block is the Alarm Management Interface (AMI) that performs data alignment by translating each alarm message toward a standard alarm message format. This is necessary because IDSs from different vendors usually produce messages according to different formats. In our implementation, we used the IDMEF format because it has been proposed as standard format by the IETF [7]. The second block, i.e. the *classification module*, is designed to label an alarm message as belonging to one or more attack classes. The *classification module* is motivated by two kinds of ambiguities: i) for a given attack, different sensors may produce a number of alarms reporting different attack descriptions; ii) an attack description may be produced by the IDS in response to different attacks. In case

of alarms labelled as belonging to more than one class, the clustering process removes the ambiguity as soon as an alarm with a unique label is clustered with the multiple-class alarm. In fact, each cluster must contain alarms belonging to one attack class. We also used the "no-class" label for those meta-alarms related to attacks for which a class has not been defined during classifier design. Details on the *training* procedure of the classification module will be given in Section 2.2. Classified alarms are sequentially sent to the *clustering/fusion* block, which applies a nearest-neighbour clustering algorithm [8]. For each received alarm the clustering algorithm determines whether the alarm can be clustered, and thus fused, to one of the clusters or have to initialize a new *meta-alarm* (a new group of alarms). The whole system is designed to be used in a near real-time environment, i.e. IDS sensors send the alarms to the alarm reduction system as soon as they are produced. In such an environment meta-alarms that have not been involved in a *clustering/fusion* process for a time interval greater than a predefined timeout threshold will be removed from the reduction system and sent in IDMEF format to the administrator.

2.1 Meta-alarm Content

Before going into the details of the clustering algorithm, let us state which information we aim to extract from each cluster, and include in the related *meta-alarm*. A *meta-alarm* is characterised by the following features: a classification name, that is the common generalized class-name assigned to the clustered alarms by the classification module; the *create-time*, that is the timestamp of the last meta-alarm update (the last fusion). Additional data include the start and stop times of the attack, list of source IP addresses, list of target IP addresses, source ports, target ports, etc. In addition, a reference to the log files of the IDSs is reported so that further investigation of the aggregated alarms can be carried out. It is worth noting that a meta-alarm M is removed from the clustering-fusion module and reported to the administrator if no more alarms are fused within a suitable time threshold.

2.2 Classification Module

An alarm class C is made up of the set of alarm messages provided by the sensors in response to attacks of type C . For example, the *portscan* alarm class is made up of the set of alarm messages obtained by simulating *portscan* attacks with different techniques. We have already noticed that a given alarm can be raised by an IDS in response to different attacks. For example, a *portscan* may cause an IDS to produce a number of alarms that refer to *DoS* attacks in addition to the alarms related to the *portscan* activity. Such *DoS* alarms are confusing because they should be produced only in case of real *DoS* attacks. The role of the classification module is to assign each alarm to the attack class(es) that might have produced it. To this end, the classifier is designed by simulating a number of real attacks for each class of attacks. For example, if we consider attacks belonging to *portscan*, *webscan* and *DoS* classes, the designing process has to be performed in the following way:

1. Simulate the most frequent *portscan* attacks with different techniques.
2. Extract the pairs {sensor-name, alarm-message} from each alarm produced by step 1.
3. Store the pairs {sensor-name, alarm-message} into a set called *portscan-descriptions*.
4. Repeat steps 1, 2 and 3 for *webscan* and *DoS* attacks, thus storing the pairs {sensor-name, alarm-message} into *webscan-descriptions* set and *dos-descriptions* set respectively.

When the classifier receives an alarm with description *Desc-1* produced by *Sensor-A*, the pair {*Sensor-A*, *Desc-1*} is compared to each pair contained into the *portscan-descriptions*, *webscan-descriptions* and *dos-descriptions* sets. The alarm is then labelled with the classes with matching pairs. For example, if the pair {*Sensor-A*, *Desc-1*} is found both into the *portscan-descriptions* set, and into the *dos-description* set, then the alarm will be labelled as belonging to both *portscan* and *DoS* classes. On the other hand, if the pair {*Sensor-A*, *Desc-1*} is not in any sets of descriptions, the alarm will be labelled as "no-class".

2.3 Clustering/Fusion Module

The clustering/fusion module is initialised by creating a number of empty sets, each one devoted to contain clusters related to one of the attack classes taken into account in the classifier design phase. In the case of the above example the clustering/fusion block creates three sets: *PortscanMetaAlarmSet*, *WebscanMetaAlarmSet*, and *DoSMetaAlarmSet*. In addition, two other sets are created, namely the *MultiClassMetaAlarmSet*, and the *NoClassMetaAlarmSet*. The first set is devoted to temporarily contain meta-alarms obtained by clustering alarms labelled as belonging to multiple classes, while the latter is devoted to contain meta-alarms obtained by clustering alarms that have not received a label from the classification module. It is worth recalling that alarm clustering is aimed at reducing the number of alarms produced by a certain class of attacks. Thus, a number of alarms are clearly not taken into account in the design phase. Before giving the details of the clustering algorithm, some definitions have to be introduced:

- **Definition 1.** Distance between pairs of features
Let us denote the distance between the i -th feature of an alarm A and the corresponding feature of a meta-alarm M as $dist(A.feat_i, M.feat_i)$. Distance measures for various types of features such as the timestamp, target IP, target port, source IP, source port, etc., are defined in Section 2.4.
- **Definition 2.** Clustering function
An alarm A is assigned to the nearest cluster if the distance between A and the meta-alarm M associated with that cluster is below a predefined threshold. In particular, alarm A is assigned to the nearest cluster M if all the distances between the corresponding features are smaller than a set of predefined thresholds:

$$\text{dist}(A.\text{feat}_i, M.\text{feat}_i) \leq \text{thres}_i \quad \forall i = 1, \dots, v \quad (1)$$

where v is the total number of features. The values of the thresholds $\{\text{thres}_i\}_{i=1..v}$ depend on the class M belongs to, as well as on the characteristics of the protected network. In the following, we will refer to an alarm A and a meta-alarm M satisfying Eq. 1, to be correlated.

– **Definition 3.** Distance between an alarm and a meta-alarm

If an alarm A and a meta-alarm M are correlated, then their distance is computed as the time gap between the create-time of A and the create-time of the more recent alarm fused to M . Otherwise, the distance between A and M is set to $+\infty$.

– **Definition 4.** Distance between an alarm and a meta-alarm set

The distance between an alarm A and a meta-alarm set S is defined in the following way:

1. If S does not contain any meta-alarm M correlated to A , then the distance is set to $+\infty$.
2. If S contains k meta-alarms M_1, M_2, \dots, M_k correlated to A , the distance between A and S is computed as $\min_{i=1..k}(\text{dist}(A, M_i))$.

In order to explain how the proposed clustering algorithm works, let us resort to an example. Let us suppose to be in a running state, and that each meta-alarm set contains a number of clusters. When a new alarm A is processed by the clustering module, three different cases may occur:

a) A has been labelled as belonging to a unique class.

If the alarm A has been labelled, for example, as a *portscan* the following distances will be computed:

$$\begin{aligned} d_1 &= \text{dist}(A, \text{PortscanMetaAlarmSet}) \\ d_2 &= \text{dist}(A, \text{MultiClassMetaAlarmSet}) \\ d_3 &= \text{dist}(A, \text{NoClassMetaAlarmSet}) \end{aligned}$$

If $(d_1 = d_2 = d_3 = +\infty)$, then there is no meta-alarm correlated to A into the *Portscan*, *MultiClass*, and *NoClass* meta-alarm sets. In this case, A will be inserted into the *PortscanMetaAlarmSet* where it will initialize a new meta-alarm. If $d_1 = \min\{d_1, d_2, d_3\}$, A will be inserted into the *PortscanMetaAlarmSet*, and it will be fused with the nearest portscan meta-alarm that is correlated to A . Similarly, if d_2 or d_3 exhibit the minimum distance, A will be inserted respectively into the *MultiClassMetaAlarmSet* or the *NoClassMetaAlarmSet*, and it will be fused with the nearest correlated meta-alarm. In the case of $d_2 = \min\{d_1, d_2, d_3\}$, the resulting meta-alarm will be moved from the *MultiClassMetaAlarmSet* to the *PortscanMetaAlarmSet*, as the alarm A has a unique class label that can resolve the ambiguity of the correlated meta-alarm. In the case of $d_3 = \min\{d_1, d_2, d_3\}$, the class label given to A will not be further considered, and the resulting meta-alarm will have no class label. The reason for computing the distances d_2 and d_3 instead of immediately insert A into *PortscanMetaAlarmSet* (A has been

labeled as a *portscan* by the classification module) is justified by the following considerations: 1) Let us assume that alarm A is the n -th alarm caused by a *portscan* attack, and that the first $n - 1$ alarms have been classified as belonging to multiple classes, *portscan* class included. By comparing A with the meta-alarms contained in the *MultiClassMetaAlarmSet*, it will be correctly fused with the correct sequence of alarms. 2) Given that a perfect matching is required among the features of the alarm A and those of a no-class meta-alarm M to be correlated (Eq. 1), if $d_3 = \min\{d_1, d_2, d_3\}$, A and M are quite certainly related to the same attack even though A has been labelled as belonging to a certain class.

- b) A has been labelled as belonging to multiple classes.

If alarm A has been labelled, e.g. as *portscan* and *DoS*, the following four distances will be computed:

$$\begin{aligned} d_1 &= \text{dist}(A, \text{PortscanMetaAlarmSet}) \\ d_2 &= \text{dist}(A, \text{DosMetaAlarmSet}) \\ d_3 &= \text{dist}(A, \text{MultiClassMetaAlarmSet}) \\ d_4 &= \text{dist}(A, \text{NoClassMetaAlarmSet}) \end{aligned}$$

if ($d_1 = d_2 = d_3 = d_4 = +\infty$), A will be inserted into the *MultiClassMetaAlarmSet*, and it will initialize a new meta-alarm. If one or more distances are not equal to $+\infty$, then A will be inserted into the nearest meta-alarm set, and it will be fused with the nearest meta-alarm.

- c) A has been labelled as belonging to none of the classes.

If A has been labelled as belonging to no-class, then it will be inserted into the *NoClassMetaAlarmSet*, and it will be clustered with the nearest no-class meta-alarm. If the *NoClassMetaAlarmSet* contains no meta-alarm correlated to A , then A will initialize a new no-class meta-alarm that inherits A 's features. It is worth recalling that an alarm A and a no-class meta-alarm M are considered correlated only if all A 's and M 's features (except the attack description) perfectly match. In this case there is a high probability that A and M are relative to the same attack, even if the attack descriptions do not coincide.

2.4 Distances Among Features

In this section we present the definition of some of the distances among features used by the clustering algorithm. Let A be an alarm and M a meta-alarm. Distances among IP addresses and port lists hold the same definitions either they refer to target or source information (i.e. $\text{dist}(A.\text{sourceIP}, M.\text{sourceIP})$ and $\text{dist}(A.\text{targetIP}, M.\text{targetIP})$ have the same definition, as well as distances among source or target port lists).

- $\text{dist}(A.IP, M.IP)$: We consider only IPv4 addresses. The distance is defined as a sub-network distance. We take the binary expression of $A.IP$ and $M.IP$, then we XOR the two binary strings. If we call n the number of zeros in the resulting binary string counted starting from the right, the distance d will be $d = 32 - n$. The greater d , the greater the distance between IP addresses.

- $dist(A.portList, M.portList)$: The distance among $A.portList$ and $M.portList$ equals the number of port numbers present in $A.portList$ but not in $M.portList$.
- $time_distance(A, M)$: The time distance t among an alarm A and a meta-alarm M is computed as the distance, in terms of milliseconds, among $A.createTime$ and $M.stopTime$.

3 Experiments

The proposed alarm clustering strategy has been implemented and tested on a live network containing dozens of hosts, some of them chosen as victims. It is worth noting that at present no dataset is publicly available for designing and testing alarm clustering algorithms. As a consequence, a direct comparison with results in the literature is rather difficult. Thus, researchers usually evaluate their algorithms by performing some experiments on a typical network scenario, and assessing the effectiveness of the proposed techniques on such a scenario. The traffic of the considered network was made up of the so-called background traffic, i.e., the normal activity of the users of the network, and by a number of simulated attacks. Three IDSs have been used to monitor network traffic: Snort 2.1.0 [9], Prelude-NIDS 0.8.6 [10], and ISS Real Secure Network Sensor 7.0 [11]. We have subdivided the experiments into three stages: 1) Training of the classification module; 2) Tuning of the thresholds involved in the clustering algorithm; 3) Performance tests. The first two stages have been carried out by executing attack simulations in an isolated network made up of three hosts, two victims (a Linux host and a Win2k host), and an attacker host. The performed experiments were related to three attack classes, i.e., *portscan*, *webscan*, and *DoS*, as they usually produce a large number of alarms.

3.1 Training and Tuning

The classification module has been designed using a number of tools available in the Internet. In particular, we have used *nmap*, *wups*, etc., as *portscan* tools; *nikto*, *babelweb*, etc., as *webscan* tools; *teardrop*, *jolt* (aka ping of death), *synflood*, etc., as *DoS* attacks. During this phase, for each attack, the pairs $\{sensor_name, alarm_message\}$ has been stored according to the procedure described in section 2.2. The values of the thresholds used by the clustering algorithm described in Section 3.3 have been estimated in two phases. In the first phase, an initial value for the thresholds has been chosen by heuristics based on attack characteristics. Then, in the second phase, attack simulations have been performed in the

Table 1. Values of the thresholds used in the clustering algorithm

Meta-Alarm Class	SourceIP	TargetIP	SourcePort	TargetPort	Time(s)
<i>portscan</i>	0	0	$+\infty$	$+\infty$	$480+\delta t$
<i>webscan</i>	0	0	$+\infty$	0	$120+\delta t$
<i>DoS</i>	$+\infty$	0	$+\infty$	$+\infty$	$120+\delta t$
<i>no-class</i>	0	0	0	0	$0+\delta t$

isolated network to suitably tune the thresholds in order to effectively cluster all the correlated alarms produced by a given attack. The notion of effectiveness may change according to the characteristic of the protected network, the needs of the network administrator, etc. Thus different tunings may fit different administrator’s needs. The thresholds used in our experiments are reported in Tab.1. The δt constant in the Time threshold column accounts for possible drifts among IDS sensors’ clock. In our experiments, δt was set equal to one second.

3.2 Performance Tests

A large number of attacks have been executed in the selected live network to test the feasibility of the designed system to correctly cluster attacks. Results showed that the proposed technique produced not only meta-alarms related to the simulated attacks, but also meta-alarms related to the background traffic. As the design phase has been carried out in an isolated network, this results show the feasibility of the proposed approach. Table 2 reports the details of the most significant results.

Portscan - When portscans have been performed, the clustering algorithm successfully produced a meta-alarm for every portscan activity. As an example, consecutive SYN and Xmas portscans have been performed from one source towards a victim, producing a total of 3074 alarms from the three considered IDSs (see the first column in table 2). During these attacks, the sensors also produced alarms related to malicious activities in the background traffic. The clustering algorithm correctly produced one meta-alarm related to the portscan, and 15 meta-alarms related to other activities. The meta-alarm related to portscan activities is correctly labelled as portscan, and contains the list of scanned ports, the source and target hosts, the start and stop times of the attack, and references to the alarms that originated the meta-alarm.

Webscans - Similar results have been also attained with webscans. In some cases, long attacks originated more than one meta-alarm, because of time gaps among groups of alarms. This kind of anomalies can be resolved by a post-

Table 2. Experimental results on live network

<i>Attack Type</i>	portscan	webscan1	webscan2	DoS
<i>Alarms from Snort</i>	1058	94	7143	71
<i>Alarms from Prelude</i>	1314	93	6601	8828
<i>Alarms from ISS Real Secure</i>	728	63	5586	186
<i>Total Number of Alarms</i>	<i>3100</i>	<i>250</i>	<i>19330</i>	<i>9085</i>
<i>Attack-related Alarms</i>	3074	244	19164	9028
<i>Alarms produced by backgroud traffic</i>	26	6	166	57
<i>Meta-Alarms from simulated attacks</i>	1	1	37	4
<i>Meta-Alarms from backgroud traffic</i>	15	3	85	33
<i>Total number of Meta-Alarms</i>	<i>16</i>	<i>4</i>	<i>122</i>	<i>37</i>

processing situation refinement module that is aimed at finding relationships among meta-alarms. As an example, Webscan2 (nikto) originated 19164 alarms that were clustered into 37 clusters. The size of the first two clusters was equal to 7464 and 11631 alarms, respectively. It is worth noting that 69 alarms produced by the ISS Real Secure sensor generated 35 meta-alarms. These alarms were related to attack responses produced by the webserver that were not correctly recognized by Real Secure.

DoS - Four DoS attacks have been performed against the same host. The proposed alarm clustering was able to correctly produce 4 meta-alarms corresponding to the different attacks carried out and 33 meta-alarms corresponding to alarms related to suspicious background-traffic.

4 Conclusions

In this paper we proposed a novel on-line alarm-clustering algorithm whose main objective is the reduction of the volume of alarms produced by today's IDS sensors. The clustering algorithm has been devised to work in near real time. Experiments performed in different attack scenarios on a live network showed that the proposed algorithm effectively groups alarms related to the same attack, even though IDSs produced alarms whose descriptions were erroneously referred to different types of attacks. The produced meta-alarms provide the system administrator with a concise high-level description of the attack. In addition, it is the starting point for the development of modules for situation refinement and threat analysis.

References

1. J. Haines, D. K. Ryder, L. Tinnel, S. Taylor, *Validation of Sensor Alert Correlators*, IEEE Security Privacy, January-February 2003, 1(1), pp. 46-56.
2. A. Valdes, K. Skinner, *Probabilistic Alert Correlation*, RAID 2001. LNCS 2212, pp. 54-68.
3. F. Cuppens, *Managing Alerts in a Multi-Intrusion Detection Environment*, Proceedings of ACSAC'01, IEEE Computer Society.
4. F. Cuppens, A. Mige, *Alert Correlation in a Cooperative Intrusion Detection Framework*, Proceedings of the IEEE Symposium on Security and Privacy, 2002.
5. P. A. Porras, M. W. Fong, A. Valdes, *A Mission-Impact-Based Approach to INFOSEC Alarm Correlation*, RAID 2002. Springer-Verlag, LNCS 2516, pp. 95-114.
6. J. Undercoffer, A. Joshi, J. Pinkston, *Modeling Computer Attacks: An Ontology for Intrusion Detection*, RAID 2003. Springer-Verlag, LNCS 2820, pp. 113-135.
7. D. Curry, H. Debar, B. Feinstein, *The Intrusion Detection Message Exchange Format* (<http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-11.txt>)
8. A.K. Jain, M.N. Murty, P.J. Flynn, *Data clustering: a review*, ACM Computing Surveys 31(3) 1999, 264-323.
9. *Snort, Lightweight Intrusion Detection for Networks*. (<http://www.snort.org>)
10. *Prelude Intrusion Detection System*. (<http://www.prelude-ids.org>)
11. *ISS, Inc.: RealSecure intrusion detection system*. (<http://www.iss.net>)

Clustering Document Images Using Graph Summaries

Eugen Barbu, Pierre Hérroux, Sébastien Adam, and Eric Trupin

Laboratoire Perception - Systèmes - Information,
FRE CNRS 2645, Université de Rouen,
UFR des Sciences & Techniques,
Place Emile Blondel,
76821 Mont-Saint-Aignan Cedex, France
Eugen.Barbu@Univ-Rouen.Fr

Abstract. Document image classification is an important step in document image analysis. Based on classification results we can tackle other tasks such as indexation, understanding or navigation in document collections. Using a document representation and an unsupervised classification method, we can group documents that from the user point of view constitute valid clusters. The semantic gap between a domain independent document representation and the user implicit representation can lead to unsatisfactory results. In this paper we describe document images based on frequent occurring symbols. This document description is created in an unsupervised manner and can be related to the domain knowledge. Using data mining techniques applied to a graph based document representation we found frequent and maximal subgraphs. For each document image, we construct a bag containing the frequent subgraphs found in it. This bag of “symbols” represents the description of a document. We present results obtained on a corpus of graphical document images.

1 Introduction

A document image analysis (DIA) system transforms a document image into a description of the set of objects that constitutes the information on the document and which are in a format that can be further processed and interpreted by a computer [1]. Documents can be classified in mostly graphical or mostly textual documents [2]. The mostly textual documents also known as structured documents respect a certain layout and powerful relations exist between components. Examples of such documents are technical papers, simple text, newspapers, program, listing, forms,...

Mostly graphical documents do not have strong layout restrictions but usually relations exist between different document parts. Examples of this type of documents are maps, electronic schemas, architectural plans...

For these two categories of documents, graph based representations can be used to describe the image content (e.g. region adjacency graph [3] for graphical and Voronoi-based neighborhood graph [4] for textual document images).

In this paper we present an approach similar to the “bag of words” method used in Information Retrieval (IR) field. We describe a document using a bag of symbols found automatically using graph mining [5] techniques. In other words, we consider the frequent subgraphs of a graph-based document representation as “symbols” and we investigate whether the description of a document as a bag of “symbols” can be profitably used in a classification task.

The approach has the ability to process document images without knowledge of, or models for, document content. In the literature one can find papers dealing with representations of textual documents using frequent items [6] and description of XML documents using frequent trees [7] but we do not know of any similar approaches in the DIA field.

The motivation for our study is the fact that unsupervised classification can represent the starting point for semi-supervised classification or indexation and retrieval from document collections. Also, the existing clustering solutions for document images are usually domain dependent and can not be used in an “incoming document flux” (fax, business mail,...) setting, where supervised techniques are not at hand.

The outline of this paper is as follows. In section 2 we present a graph representation and how we create this representation from a document image. Section 3 presents the graph-mining method used, in section 4 we describe how we create clusters based on dissimilarities between bags of symbols. Section 5 presents some experimental results. We conclude the paper and outline perspectives in section 6.

2 Document Graph Based Representations

Eight levels of representation in document images are proposed in [8]. These levels are ordered in accordance with their aggregation relations. Data array level, primitive, lexical, primitive region, functional region, page, document, and corpus level are the representation levels proposed.

Without losing generality, in the following paragraphs we focus our attention on a graph-based representation build from the primitive level. The primitive level contains objects such as connected components (sets of adjacent pixels with the same color) and the relations between them.

Let I be an image and $C(I)$ the connected components from I , if $c \in C(I)$, c is described as $c=(id,P)$, where id is a unique identifier and P the set of pixels the component contains. Based on this set P , we can compute the center for the connected component bounding box and also we can associate a feature vector to it. Based on that, $c=(id,x,y,v), v \in R^n$. Subsequently using a clustering procedure on the feature vectors we can label the connected component and reach the description $c=(id,x,y,l)$ where l is a nominal label. The graph $G(I)$ representing the image is $G=G(V(I),E(I))$. Vertices $V(I)$ correspond to connected components and are labeled with component labels. An edge between vertex u and vertex w exists iff $((u.x-w.x)^2+(u.y-w.y)^2)^{1/2} < t$, where t is a threshold that depends on the image I global characteristics (size, number of connected components,...).

The exact methodology employed to construct the graph representation is subsequently presented. From a binary document image we extract connected components (black and white). The connected components will be the graph nodes. For each connected component we extract features. In the actual implementation the extracted characteristics are rotation and translation invariant features based on Zernike moments [9]. The invariants represent the magnitudes of a set of orthogonal complex moments of a normalized image.

The following step is to associate each connected component a label.

2.1 Labeling Connected Components

The two main categories of clustering methods are partitional and hierarchical. Partitional methods can deal with large sets of objects (“small” in this context means less than 300) but needs the expected number of clusters in input. Hierarchical methods can overcome the problem of number of clusters by using a stopping criterion [10] but are not applicable on large sets due to their time and memory consumption.

In our case the number of connected components that are to be labeled can be larger than the limit of applicability for hierarchical clustering methods. In the same time we cannot use a partitional method because we do not know the expected number of clusters. Based on the hypothesis that a “small” sample can be informative for the geometry of data, we obtain in a first step an estimation for the number of clusters in data. This estimation is made using an ascendant clustering algorithm with a stopping criterion. The number of clusters found in the sample is used as input for a partitional algorithm applied on all data.

We tested this “number of cluster estimation” approach using a hierarchical ascendant clustering algorithm [10] that uses Euclidean distance to compute the dissimilarity matrix, complete-linkage to compute between-clusters distances, and Calinsky-Harabasz index [11] as a stopping criterion. The datasets (T_1, T_2, T_3) (see Table 1) are synthetically generated and contain well separated (not necessarily convex) clusters.

Table 1. Data sets description

T	T	no. of clusters
T1	24830	5
T2	32882	15
T3	37346	24

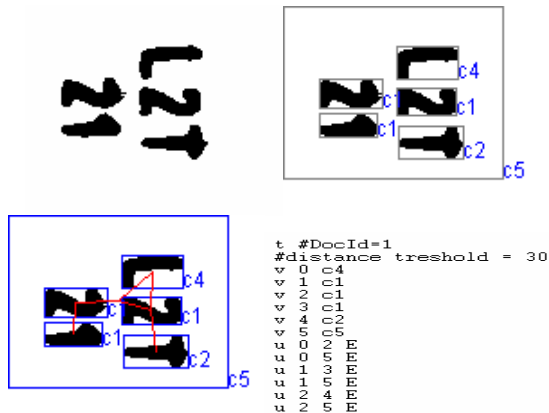
Considering S the sample extracted at random from a test set, in Table 2 we present predicted cluster numbers obtained for different sample sizes. After repeating the sampling procedure for 10 times if the test set is for example $|S|=50$, we obtain a set of estimations for the number of clusters. We can see that by using a majority voting decision rule we can find the good number of clusters in most of the cases and even when the sample size is very small (50 or 100) compared to the data set size.

Table 2. Proposed number of clusters

T \ S	50	100	300	500	600	700
T1	[6, 8, 7, 6, 5, 6, 6, 6, 5, 5] 6	[5, 7, 9, 7, 5, 5, 7, 5, 5, 7] 5	[7, 5, 7, 8, 7, 5, 5, 5, 7, 7] 7	[8, 7, 5, 5, 5, 5, 5, 5, 5, 5] 5	[5, 5, 5, 5, 5, 7, 7, 7, 7, 5] 5	[5, 5, 7, 5, 7, 5, 5, 7, 5, 5] 5
T2	[9, 15, 15, 14, 13, 15, 13, 13, 14, 15] 15	[15, 15, 13, 15, 15, 15, 15, 15, 15, 15] 15	[15, 15, 15, 15, 15, 15, 15, 15, 15, 14] 15	[15, 15, 15, 15, 15, 15, 15, 15, 15, 15] 15	[15, 15, 15, 15, 15, 15, 15, 15, 15, 15] 15	[15, 15, 15, 15, 15, 15, 15, 15, 14, 15] 15
T3	[11, 7, 9, 18, 7, 7, 6, 4, 14, 8] 7	[6, 14, 23, 21, 7, 17, 23, 16, 12, 11] 23	[22, 24, 23, 19, 23, 24, 24, 24, 21, 21,24,] 24] 24	[21, 25, 25, 24, 22, 24, 23, 24, 24, 24] 24	[20, 25, 21, 24, 19, 23, 24, 25, 24, 22] 24	[23, 20, 21, 20, 25, 24, 24, 21, 25, 24] 24

We employed our sampling approach combined with the k-medoids clustering algorithm [12] on the connected components data set from images in our corpus (see section 5). The k-medoids clustering algorithm is a more robust version of the well known k-means algorithm. The images from our corpus contain 6730 connected components. The proposed number of clusters using ten samples of size 600 is [16,14,17,16,16,19,7,17,15,16] and by considering the majority we use 16 clusters as input to the partitional clustering algorithm.

After labeling the connected components (nodes in the graph) subsequently we describe the way we add edges to the graph. The edges can be labeled or not (if unlabeled the significance is Boolean: we have or have not a relation between two connected components) and can be relations of spatial proximity, based on “forces” [13], orientation or another criterion. In our actual implementation the distance between centers of connected components is used (see Fig. 1). If the distance between two connected components centers is smaller than a threshold, then an edge will link the two components (nodes).

**Fig. 1.** Image and its associated graph

3 Graph Mining

The main objective of graph mining is to provide new principles and efficient algorithms to mine topological substructures embedded in graph data” [5].

Mining frequent patterns in a set of transaction graphs is the problem of finding in this set of graphs those subgraphs that occur more times in the transactions than a threshold (minimum support). Because the number of patterns can be exponential this problem complexity can also be exponential. An approach to solve this problem is to start with finding all frequent patterns with one element, then all patterns with two elements, etc in a level-by-level setting. In order to reduce the complexity different constraints are used: the minimum support, the subgraphs are connected, and not overlapped. An important concept is that of maximal subgraph. A graph is said to be maximal if it does not have a frequent super-graph. In our document image analysis context we are interested in finding maximal frequent subgraphs because we want to find symbols but to ignore their parts.

A system that is used to find frequent patterns in graphs is FSG (Frequent Subgraph Discovery) that “finds patterns corresponding to connected undirected subgraphs in an undirected graph database”[14]. The input for the FSG program is a list of graphs and a minimum support threshold. Each graph represents a transaction. We present subsequently how we construct the transaction list starting from a set of document images. Using the procedure presented in section 2 we create for each document an undirected labeled graph. Every connected component of this graph represents a transaction. Using FSG we extract the frequent subgraphs and we construct a bag of frequent subgraphs occurring in each document.

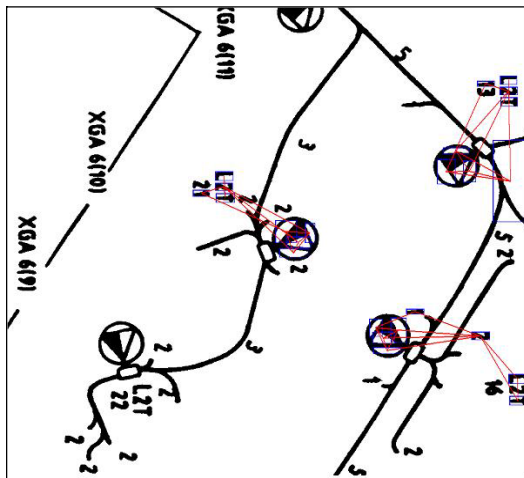


Fig. 2. Frequent subgraph and its occurrences in an image

In the following paragraphs we consider that the frequency condition is sufficient for a group of connected components to form a symbol and we will conventionally

make an equivalence between the frequent subgraphs found and symbols. As we can see in the example (Fig. 2) the proposed symbols are far from being perfect due to the image noise, connected components clustering procedure imperfections, ... however we can notice the correlation between this artificial symbol and the domain symbols.

4 Documents Description

A document can be seen as a bag of symbols $A=(s_1,s_1,s_1,s_2,s_2,s_3,\dots,s_n)$. We can use this representation as it is but we can also apply a weighting schema on it in order to distinguish between symbols with different discriminative power.

A collection of documents is represented by a symbol-by-document matrix A , where each entry represents the occurrences of a symbol in a document image, $A=(a_{ik})$, where a_{ik} is the weight of symbol i in document k . Let f_{ik} be the number of occurrences of symbol i in document k , N the number of documents in the collection, and n_i the total number of times symbol i occurs in the whole collection. In this setting conform with [15] one of the most effective weighting scheme is entropy-weighting. The weight for symbol i in document k is given by :

$$a_{ik}=\log(1+f_{ik})*(1+\frac{1}{\log(N)}\sum_{j=1}^N\frac{f_{ij}}{n_i}\log(\frac{f_{ij}}{n_i}))$$

Now, considering two documents A, B with the associated weights $A=(a_1,a_2,\dots,a_t), B=(b_1,b_2,\dots,b_t)$ where t is the total number of symbols, then

$$d(A,B)=1-\frac{\sum_{i=1}^t a_i*b_i}{(\sum_{i=1}^t a_i^2 \sum_{i=1}^t b_i^2)^{1/2}}$$

represents a dissimilarity measure based on the cosine correlation.

5 Experimental Results

A comparison between results obtained using the proposed document representation and three other representations is made in the following paragraphs. On a corpus of graphical document images we have extracted different sets of features. Each document image is described with one of the following types of features : Zernike moments for the whole image (a vector with 16 components) abbreviated as ZM in Table 3, pixel densities (the feature vector considered is composed of the 85 (1+4+16+64) gray levels of a 4-level-resolution pyramid [16] , see Fig 3.), (QT), weighted connected components label list , and symbol label list .

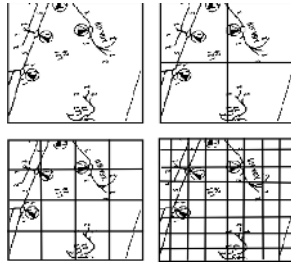


Fig. 3. Four level resolution pyramid

Using a hierarchical ascendant clustering procedure on the dissimilarities between document representations (as Zernike moments, pixels densities, ...) combined with Calinsky-Harabasz stopping criterion we obtain four partitions that were compared with the ground-truth partition of the corpus.

In order to evaluate the partitions proposed by the clustering algorithm, we employ the overall F-measure index. Let D represent the set of documents and let $C = \{C_1, \dots, C_k\}$ be a clustering of D . Also let $C' = \{C'_1, \dots, C'_l\}$ the reference (ground truth) classification.

Then the recall of cluster j with respect to class i is $rec(i, j) = \frac{|C_j \cap C_i|}{C_i}$,

the precision $prec(i, j) = \frac{|C_j \cap C_i|}{C_j}$ and F-measure $F_{ij} = \frac{2prec(i, j)rec(i, j)}{prec(i, j) + rec(i, j)}$. The

overall F-measure of a clustering is: $F = \sum_{i=1}^l \frac{|C_i|}{|D|} * \max\{F_{ij}\}_{j=1..k}$. F-measure is 1.0 if

the matching between the two partitions (ground truth and the one proposed by the clustering algorithm) is perfect.

Our corpus contains 30 images from the class of a French telephony operator (FT) maps, 25 electronic schemas, and 5 architectural plans.

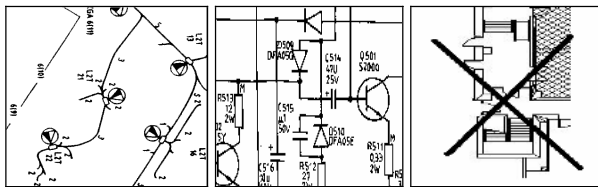


Fig. 4. Corpus images

This images are scanned images that contains real and artificial noise.

We can see that the connected component list approach obtains good results compared with the simple approaches (Zernike moments and densities). In the same time the symbols list approach representation is more compact than the connected components list and obtains better results.

Table 3. Results on our corpus

	ZM	Densities	Connected Components list	Symbols list																														
F-measure	0.58	0.69	0.89	0.90																														
Confusion matrix	<table border="1"><tr><td>1</td><td>29</td></tr><tr><td>0</td><td>25</td></tr><tr><td>0</td><td>5</td></tr></table>	1	29	0	25	0	5	<table border="1"><tr><td>30</td><td>0</td></tr><tr><td>25</td><td>0</td></tr><tr><td>0</td><td>5</td></tr></table>	30	0	25	0	0	5	<table border="1"><tr><td>26</td><td>4</td><td>0</td></tr><tr><td>2</td><td>1</td><td>22</td></tr><tr><td>0</td><td>5</td><td>0</td></tr></table>	26	4	0	2	1	22	0	5	0	<table border="1"><tr><td>26</td><td>4</td><td>0</td></tr><tr><td>0</td><td>3</td><td>22</td></tr><tr><td>0</td><td>5</td><td>0</td></tr></table>	26	4	0	0	3	22	0	5	0
	1	29																																
	0	25																																
0	5																																	
30	0																																	
25	0																																	
0	5																																	
26	4	0																																
2	1	22																																
0	5	0																																
26	4	0																																
0	3	22																																
0	5	0																																

Table 4. How to read the confusion matrix

Cluster 1	Cluster 2	
1	29	←FT maps
0	25	←Electronic schemas
0	5	←Architectural drawings

6 Conclusions

The research undertaken represents a novel approach for clustering document images. The approach uses data mining tools for knowledge extraction. It automatically finds frequent symbols. These frequent patterns are part of the document model and can be put in relation with the domain knowledge. The exposed method can be applied to other graph representations of a document. In the near future, we will apply this approach to layout structures of textual document images.

Another follow up activity is to quantify the way noise can affect the connected components labeling, and the manner in which the inexact number of clusters can affect the graph mining procedure. Based on this error propagation study we can ameliorate our method.

Other possible improvements can be obtained if we would employ a graph-based technique that can deal with error tolerant graph matching.

References

1. Antonacopoulos A. Introduction to Document Image Analysis, 1996.
2. Nagy G. Twenty years of document analysis in PAMI. IEEE PAMI, 22:38-62, 2000.
3. Pavlidis, T., Algorithms or Graphics and Image Processing, Computer Science Press, 1982
4. Bagdanov A.D. and M. Worring, "Fine-grained Document Genre Classification Using First Order Random Graphs", ICDAR 2001,79-90.
5. Washio T., Motoda H., State of the art of graph-based data mining. SIGKDD Explor. Newsl.vol. 5, no 1,pp. 59-68 ,2003.

6. Fung, B. C. M., Wang, K., & Ester M. Hierarchical Document Clustering Using Frequent Itemsets. Proceedings of the SIAM International Conference on Data Mining, 2003.
7. Termier A., Rousset M., and Sebag M., "Mining XML Data with Frequent Trees", DBFusion Workshop'02, pages 87-96, 2002.
8. Blostein D., Zanibbi R., Nagy G., and Harrap R., "Document Representations", GREC 2003
9. Khotazad A., and Hong Y.H., "Invariant Image recognition by Zernike Moments", IEEE PAMI, Vol 12, No 5, May 1990
10. Gordon A.D. "Classification 2nd Edition", 1999.
11. Milligan, G. W., Cooper, M.C.: An Examination of Procedures for Determining the Number of Clusters in a Data Set. *Psychometrika*, 58(2), (1985) 159-179.
12. L. Kaufmann and P. J. Rousseeuw. Clustering by means of medoids. In *Statistical Data Analysis based on the L 1 Norm and Related Methods*, pages 405—416, 1987.
13. Salvatore Tabbone, Laurent Wendling, Karl Tombre, "Matching of graphical symbols in line-drawing images using angular signature information" *Int'l Journal on Document Analysis and Recognition*, Vol. 6, No. 2, 2003, 115-125.
14. Seno M., Kuramochi M., and Karypis G., PAFI, A Pattern Finding Toolkit, <http://www.cs.umn.edu/~karypis>, 2003.
15. Dumais, S.T. , Improving the retrieval information from external resources, *Behaviour Research Methods, Instruments and Computers*, Vol. 23, No. 2, pp. 229-236, 1991.
16. Ballard D.H., Brown C.M., "Computer Vision", Prentice Hall, 1982

Feature Selection Method Using Preferences Aggregation

Gaëlle Legrand and Nicolas Nicoloyannis

Laboratoire ERIC, Université Lumière Lyon 2,
Bât. L-5, av. Pierre Mendès-France,
69676 Bron Cedex – France
glegrand@eric.univ-lyon2.fr
nicolas.nicoloyannis@univ-lyon2.fr

Abstract. The feature selection allows to choose P features among M ($P < M$) and thus to reduce the representation space of data. This process is increasingly useful because of the databases size increase. Therefore we propose a method based on preferences aggregation. It is an hybrid method between filter and wrapper approaches.

1 Introduction

Due to increasing size of databases, the improvement of data representation quality becomes a main problem in data mining. One of the major difficulties related to data representation quality is data dimension. This problem is linked with the number of exogenous features characterizing each object. Users who want to cover all existing aspects of an endogenous feature and to obtain comprehensible knowledge define a great number of exogenous features. However, among these features, some will be irrelevant, useless and/or redundant. Indeed, it is often difficult or even impossible to distinguish the relevant features from the irrelevant ones.

The problem of data dimension can be summarized by "Less is more" from Liu and Motoda [21] which means that if we wish to extract useful and comprehensible information from our data, it is initially appropriate to delete irrelevant parts. Feature selection solves this problem. It chooses an optimal features subset according to a particular criterion and reduces the features space by removing those which are irrelevant. Feature selection eliminates useless and redundant features, the learning process is then accelerated and the accuracy of learning algorithms may be improved. It also permits to reduce noise generated by some features. There are a lot of feature selection methods which are gathered in two approaches: the wrapper approach, [10], which use the learning algorithm to test all existing features subsets, and the filter approach, [12], which corresponds to a data pre-processing step preceding the learning phase. The fundamental difference between these two families lies in the fact that the first is related to the learning algorithm whereas the second is completely independent of it.

1.1 Wrapper Methods

These methods [5] take the influence of the selected features subset on the performances of the learning algorithm into account. The learning algorithm is used as an evaluation function to test different features subsets. However, its computational cost is too important in most cases [17] : these methods generate all existing features subset.

1.2 Filter Methods

Filter approaches are grouped into 5 categories : complete, heuristic, random, fast sequential selection and step by step.

Complete Methods test all possible subsets of P features among M features with M the total number of features and P the number of selected features. We can quote MDLM [32] or FOCUS [1], [2] or PRESET [26]. MDLM performs a comparison of all existing features subsets. PRESET is an algorithm based on the rough sets theory. It selects a features subset, named reduction which involves the same consistency on the learning set as the initial features set. All features not belonging to this reduction are eliminated. FOCUS makes a complete search among all features subsets and selects the minimal subset which allows to determine the class of each object. The complexity of FOCUS is about $O(N^M)$, with N the number of objects and M the number of features. These 3 algorithms are impossible to apply in most of cases due to their very high computational cost.

Heuristic Methods have many representatives. We present only the principal ones. Relief, [13], is an iterative features weight-based algorithm inspired by instance-based learning algorithms. Relief knew many alternatives. The most interesting one is ReliefF, [15], which deals with multi-classes problems. The complexity of Relief and its alternatives is $O(IMN)$ where I is the number of iterations fixed by the user. The Branch and Bound methods, [27], use a selection criterion characterized by the monotonicity property : all subsets, for which the selection criterion is not higher than a threshold, are eliminated. ABB, [20], uses the same principle with inconsistency rate. Its complexity is $O(N2^M)$. The Khi2 Algorithm, [18] carries out simultaneously the features discretization and the elimination of irrelevant features. It is based on the χ^2 statistics. These methods require several accesses to databases.

Random Methods main representative is LVF, [19]. LVF selects the smallest features subset generated randomly and which satisfies an inconsistency criterion. Its complexity is $O(IMN)$, with I the number of subset generation. Because of its probabilistic property, the number of selected features tends towards the half of the initial features number. Its complexity is about $O(IMN)$. Like previous methods, these methods require several accesses to databases.

Fast Sequential Selection Method are iterative feature selection methods with a single access to database. The selection process is thus a stepwise process : the first step selects the feature X_1 that is the more correlated with endogenous feature Y ; the second step selects the feature that is the more partially correlated with Y with fixed

values for X_1 , and so on... In order to have a single database scan, fast correlation measures must be used (such as Kendall rank correlation coefficient, or Pearson correlation coefficient, or modified Rand coefficient,...). This kind of method is represented by MIFS [3], CFS [8], and the method proposed by Lallich and Rakotomalala [16]. These methods are fastest and quite efficient. They appear like the most interesting.

Step-by-step Methods use short-sighted criteria to select features. These methods do not take into account the interaction between features and classify features according to their discriminating capacity. This type of methods is effective and very rapid in particular on problems comprising at the same time many features and objects. Their complexity is $O(N \log N)$.

To sum up, wrapper approach and complete methods are inapplicable because of their computational cost and time complexity. Heuristic methods have difficulties with redundant features and, random methods are skewed towards a subset having a number of features about the half of the initial features number. Moreover, most of these methods require several scans of database which imply a high I/O cost. It consequently appears that fast sequential selection methods and step-by-step methods are the more attractive ones since they propose good results as well as very suitable computing cost.

We propose here a new feature selection algorithm. Our method does not belong to wrapper approach nor to filter approach. It is situated at the intersection of filter and wrapper approaches. It offers a reasonable processing time compared with pure wrapper methods. It uses preferences aggregation in the first stage to determine an ordered list of features subsets. The first stage is the filter part. The second and last stage is the wrapper part. The next section is devoted to the initial ideas. Third section deal with the feature selection method. Experimental evaluation is in section 4.

2 Starting Point

We start from the following observation : step by step methods using short-sighted criteria such as Shannon entropy are fast, inexpensive and have good results. There are 4 categories of criteria which measure various features specifications :

- **Information measures:** these measures determine the information gain from a feature. The feature which has the greatest information gain, will be preferred to the other features. We can mention Shannon entropy [31], gain ratio [30], normalized gain [11].
- **Distance measures:** they evaluate the separability of classes. They are also know as separability, divergence, or discrimination measures : Euclidian distance measure, Mantaras distance measure [7], Gini coefficient [6].
- **Dependence measures** are all correlation or association measures. They qualify the ability to predict the value of one feature from the value of another. They can be used to find the correlation between a feature and a class. If the correlation of feature X_1 with a class is higher than the correlation of feature X_2 with the same

class, then feature X_1 is preferred to X_2 . We can cite chi-squared, Tschuprow coefficient [9] and [25], and Cramer coefficient.

- **Consistency measures:** they use the Min-Features bias in selecting a features subset. The Min-Features bias prefers hypotheses definable over as few features as possible. Two objects are inconsistent if their modalities are identical and if they belong to two different classes. These measures detect redundant features. We can cite the τ of Zhou [33].

However, the use of a short-sighted method generates two problems:

- The choice of criterion is delicate: Which criterion is the most effective?
- The form of result (a list of sorted features) does not allow us to determine the optimal features subset.

The method we propose solves these two problems in the following way:

- There is no criterion better or more effective than others. Each criterion emphasizes some specific features qualities. It seems to be interesting to obtain a result which takes the opinion of different criteria into account. So to obtain this type of results, we use a method of preferences aggregation and several short-sighted criteria.
- Obtaining a sorted list of features limits the interest of the features selection method. Indeed, the question is : how can we determine the optimal size of a features subset? When we have a sorted list of features, one of the methods which seems to be effective to obtain an optimal subset is to use a wrapper approach which adds or removes iteratively elements of the sorted list. At each iteration, the learning algorithm tests if the addition or the suppression of a feature involves an improvement of error rate. However, this process is too expensive to be applied. For this reason, we parameterise the preferences aggregation method so that it doesn't provide an ordering on the features but a preordering. Also, we will not add features one by one but features subset by features subset.

3 Presentation of Our Method

Our feature selection method is at the intersection of filter and wrapper approaches, [35]. It is a Forward Selection method which makes feature classification possible with the use of short-sighted criteria. The result is a sorted list of disjoint features subsets. This method has 3 steps:

- Calculus and discretization of different criteria for each feature (filter approach),
- Application of preferences aggregation method on results obtained at the previous stage (filter approach),
- Research of the optimal features subset (wrapper approach).

3.1 Calculus and Discretization of Criteria

We let users choose the short-sighted criteria set. The only condition is : criteria must belong to each categories. For experiments, we select a set of 10 short-sighted criteria:

Shannon entropy, gain ratio, normalized gain, Mantaras distance measure, Gini coefficient, chi-squared, Tschuprow coefficient, Cramer coefficient, τ of Zhou. Each criterion for all features are calculated in parallel. The result obtained is a set constituted of 10 ordered lists in the order descending of feature relevance.

A feature is as relevant as another one even if the two features do not bring the same information. Therefore, we introduce the concept of features equivalence. In order to define this concept, we consider a set of objects $O = \{o_1, \dots, o_j, \dots, o_n\}$ described by a features set $X = \{x_1, \dots, x_i, \dots, x_p\}$ named initial features set. Given $CR = \{cr_1, \dots, cr_k, \dots, cr_{10}\}$ the set of 10 short-sighted criteria with $cr_k = \{cr_{k1}, \dots, cr_{ki}, \dots, cr_{kp}\}$, the set of the criterion k values for each feature of X . The cr_{ki} values of each criterion are normalized with the following transformation: for a feature $x_i \in X$ and a criterion $cr_k \in CR$, the normalized value of criterion is:

$$cr_{ki,N} = \frac{cr_{ki} - \text{Min}(\{cr_k\})}{\text{Max}(\{cr_k\}) - \text{Min}(\{cr_k\})} \tag{1}$$

After their normalization, these values are discretized in deciles. The discretization assigns to each feature $x_i \in X$ a rank for each criterion $cr_k \in CR$ as follows :

- For criteria which must be minimized :
 If $cr_{ki,N} \in [0;0.1[$ then $R_{ki} = 1$; If $cr_{ki,N} \in [0.1;0.2[$ then $R_{ki} = 2$; ... ; If $cr_{ki,N} \in [0.9;1]$ then $R_{ki} = 10$
- For criteria which must be maximized :
 If $cr_{ki,N} \in [0;0.1]$ then $R_{ki} = 10$; If $cr_{ki,N} \in [0.1;0.2[$ then $R_{ki} = 9$; ... ; If $cr_{ki,N} \in [0.9;1]$ then $R_{ki} = 1$

R_{ki} is the rank assigned to feature $x_i \in X$ for criterion $cr_k \in CR$. The most relevant feature has the smallest rank. Thus the equivalence concept is defined as follows : two features x_i and x_j are equivalents according to a criterion k if and only if for this criterion, they have the same rank :

$$(x_i \Leftrightarrow x_j) \Leftrightarrow R_{ki} = R_{kj} . \tag{2}$$

3.2 Aggregation of Criteria Results

For all preferences aggregation methods [23], it is appropriate to define a set of judges and a set of objects. In our case, the objects are initial features and the judges are criteria. We use the preferences aggregation method developed in [28] and [29] and

based on [22] and [24]. We don't describe in detail this method but we present its subjacent principle.

For each objects pair (x_i, x_j) , each judge states its opinion $A_k(i, j)$. A_k , the opinion of judge k is an application of $X \times X$ in $\{Pr ef, N Pr ef, EQ\}$.

Thus,

$$A_k(i, j) = Pr ef \Leftrightarrow \text{judge } k \text{ prefers } x_i \text{ to } x_j \Leftrightarrow R_{ki} < R_{kj},$$

$$A_k(i, j) = N Pr ef \Leftrightarrow \text{judge } k \text{ prefers } x_i \text{ to } x_j \Leftrightarrow R_{ki} > R_{kj},$$

$$A_k(i, j) = EQ \Leftrightarrow \text{judge } k \text{ considers } x_i \text{ and } x_j \text{ like equivalents} \Leftrightarrow R_{ki} = R_{kj}.$$

The result we wish to obtain is an opinion OP called opinion of broad preferences and which generates a preordering relation on X . OP is an application of $X \times X$ in $\{Pr ef, N Pr ef, EQ\}$.

Definition 1: The degree of agreement $\rho_{ij}(OP, A_k)$ between the advices $OP(i, j)$ and $A_k(i, j)$ is defined in table 1.

Table 1. Degree of agreement ρ_{ij}

OP / A_k	$Pr ef$	$N Pr ef$	EQ
$Pr ef$	1	0	1/2
$N Pr ef$	0	1	1/2
EQ	1/2	1/2	1

Definition 2: The degree of agreement $DA(OP, A_k)$ between the opinions OP and A_k is $DA(OP, A_k) = \sum_{(x_i, x_j) \in X} \rho_{ij}(OP, A_k)$.

Definition 3: The degree of agreement between the opinion OP and the opinion of all judges is $DA(OP) = \sum_{k=1}^{10} DA(OP, A_k)$.

Our problem consists in building an opinion OP which generates a preordering on X and which maximizes $DA(OP)$. The corresponding optimization problem is NP-hard, hence the use of a meta-heuristic. Simulated annealing method [14] is used for maximization. We choose simulated annealing because it's a rapid and easy to use method. The parameters are : the decay rate equal 0.98, the halting condition is a

number of iterations which equal $10 * |X|$. The neighbourhood of the current solution is defined as follow : a preordering $L' = \{l'_1, \dots, l'_h, \dots, l'_H\}$ is neighbour of a preordering L , $L' \subset V(L)$, if and only if L' derive from L by the movement of only one object. After the application of this aggregation method, we obtain an ordered list of disjoint features subsets $L = \{l_1, \dots, l_h, \dots, l_H\}$.

3.3 Optimal Features Subset

Until now, our method has a filter approach. At this stage, our method has a wrapper approach. The advantage of using a wrapper approach is the use of the influence of the features subset on learning algorithm performances. Detection of the optimal subset is carried out as follows : within the h^{th} iteration, the features subset $l_h \in L$ is added to the optimal features subset. The optimal features subset is the one having the smallest error rate on the learning set.

4 Experimentations

In our experiments we used 14 databases from the UCI collection [4]. The quantitative features are discretized with Fusinter method, [14]. The features selection is carried out on 30% of the initial set of objects while keeping the initial distribution

Table 2. Tests with ID3

Bases	Without selection		With selection	
	Error rate	Sd	Error rate	Sd
Austra	16.60	4.57	15.29	3.48
Breast	5.95	1.95	4.27	2.8
Cleve	18.53	8.68	21.9	8.67
CRX	14.73	5.68	15.7	3.1
German	31.86	7.53	26.14	4.87
Heart	27.05	10.29	26.32	11.04
Iono	21.37	8.39	11.73	5.59
Iris	3.73	4.57	4.73	4.74
Monks-1	25.22	8.3	25.18	7.56
Monks-2	34.91	6.79	34.89	6.71
Monks-3	1.28	1.28	3.88	2.69
Pima	26.11	5.43	24.5	5.15
Tic Tac Toe	33.43	5	25.16	6.31
Vehicle	34.24	4.96	28.75	5.44

Table 3. Tests with BN

Bases	Without selection		With selection	
	Error rate	Sd	Error rate	Sd
Austra	14.26	4.58	15.27	3.61
Breast	2.65	1.31	2.65	2.05
Cleve	21	6.63	17.77	6.14
CRX	14.67	3.14	15.69	3.99
German	23.71	6.58	23.43	4.62
Heart	17.37	7.46	17.89	7.14
Iono	6.83	5.06	7.25	5.88
Iris	6.45	7.14	2.82	4.31
Monks-1	25.22	6	25.19	4.68
Monks-2	38.94	4.14	34.92	5.11
Monks-3	3.88	2.9	3.85	3.67
Pima	21.14	5.42	22.83	5.73
Tic Tac Toe	29.61	5.15	27.83	3.92
Vehicle	34.27	5.52	33.95	4.18

Table 4. Tests with Sipina

Bases	Without selection		With selection	
	Error rate	Sd	Error rate	Sd
Austra	16.73	3.95	15.28	6.02
Breast	7.13	2.29	6.73	4.84
Cleve	21.47	8.57	31.67	10.87
CRX	16.3	6.22	17.13	6.05
German	28.14	5.5	31.71	4.51
Heart	23.16	10.04	27.89	7.82
Iono	7.73	6.95	6.88	2.58
Iris	4.64	6.17	4.64	6.17
Monks-1	20.11	4.89	25.18	3.72
Monks-2	38.24	7	34.89	8.79
Monks-3	1.79	2.58	3.87	3.09
Pima	24.3	4.46	25.05	4.36
Tic Tac Toe	20.67	3.77	26.06	7.5
Vehicle	47.26	6.24	50.58	5.63

Table 5. Number of selected features with ID3

Bases	Without selection	Our method	ReliefF	MIFS
Austra	14	1	2	13
Breast	9	3	6	9
Cleve	13	7	6	8
CRX	15	3	2	7
German	20	5	14	3
Heart	13	2	2	13
Iono	34	2	25	8
Iris	4	3	4	3
Monks-1	6	1	2	1
Monks-2	6	1	2	2
Monks-3	6	2	2	3
Pima	8	2	7	4
Tic Tac Toe	9	7	5	3
Vehicle	18	14	18	6

Table 6. Number of selected features with Naïve Bayesian

Bases	Without selection	Our method	ReliefF	MIFS
Austra	14	2	2	13
Breast	9	7	6	9
Cleve	13	5	6	8
CRX	15	5	2	7
German	20	9	14	3
Heart	13	8	2	13
Iono	34	26	25	8
Iris	4	2	4	3
Monks-1	6	1	2	1
Monks-2	6	1	2	2
Monks-3	6	2	2	3
Pima	8	5	7	4
Tic Tac Toe	9	7	5	3
Vehicle	18	12	18	6

Table 7. Number of selected features with Sipina

Bases	Without selection	Our method	ReliefF	MIFS
Austra	14	1	2	13
Breast	9	4	6	9
Cleve	13	1	6	8
CRX	15	3	2	7
German	20	1	14	3
Heart	13	2	2	13
Iono	34	26	25	8
Iris	4	2	4	3
Monks-1	6	1	2	1
Monks-2	6	1	2	2
Monks-3	6	2	2	3
Pima	8	1	7	4
Tic Tac Toe	9	3	5	3
Vehicle	18	10	18	6

Table 8. Tests with ReliefF and MIFS (ID3)

Bases	Our method		MIFS		ReliefF	
	Error rate	Sd	Error rate	Sd	Error rate	Sd
Austra	15.29	3.48	17.17	4.12	15.31	5.23
Breast	4.27	2.8	5.9	2.64	5.29	3.16
Cleve	21.9	8.67	24.68	10.27	40.54	7.77
CRX	15.7	3.1	16.12	6.7	17.54	5.88
German	26.14	4.87	27.43	5.06	30.14	6.01
Heart	26.32	11.04	28.42	9.76	27.38	9.06
Iono	11.73	5.59	15.75	8.71	11.78	3.94
Iris	4.73	4.74	4.82	6.58	3.73	4.57
Monks-1	25.18	7.56	25.20	7.71	55.52	3.34
Monks-2	34.89	6.71	34.91	6.7	34.9	8.63
Monks-3	3.88	2.69	3.86	2.86	3.88	3.34
Pima	24.5	5.15	24.87	4.83	25.05	7.69
Tic Tac Toe	25.16	6.31	30.81	7.11	30.51	5.9
Vehicle	28.75	5.44	40.62	7.39	42.25	6.52

Table 9. Tests with ReliefF and MIFS (Naïve Bayesian)

Bases	Our method		MIFS		ReliefF	
	Error rate	Sd	Error rate	Sd	Error rate	Sd
Austra	15.27	3.61	14.28	3.08	15.28	5.15
Breast	2.65	2.05	2.86	1.87	3.45	2.56
Cleve	17.77	6.14	20.52	11.34	40.67	4.33
CRX	15.69	3.99	14.66	5.7	16.53	2.8
German	23.43	4.62	26.29	3.63	30.71	4.96
Heart	17.89	7.14	17.89	10.04	21.05	10.53
Iono	7.25	5.88	5.22	4.4	9.32	6.22
Iris	2.82	4.31	4.64	6.17	6.45	7.14
Monks-1	25.19	4.68	25.20	7.18	51.9	8.2
Monks-2	34.92	5.11	34.92	6.24	34.92	6.65
Monks-3	3.85	3.67	3.86	2.87	3.85	3.85
Pima	22.83	5.73	21.33	4.3	25.04	3.41
Tic Tac Toe	27.83	3.92	28.87	5.42	27.97	4.19
Vehicle	33.95	4.18	39.85	8.01	45.82	8.78

Table 10. Tests with ReliefF and MIFS (Sipina)

Bases	Our method		MIFS		ReliefF	
	Error rate	Sd	Error rate	Sd	Error rate	Sd
Austra	15.28	6.02	16.35	6.65	15.28	5.25
Breast	6.73	4.84	7.13	2.29	5.9	3.8
Cleve	31.67	10.87	30.41	10.7	40.56	10.4
CRX	17.13	6.05	17.95	5.23	16.12	4.72
German	31.71	4.51	26.29	4.53	31	4.61
Heart	27.89	7.82	23.16	6.74	22.11	6.57
Iono	6.88	2.58	7.70	6.22	19.4	6.85
Iris	4.64	6.17	4.55	9.32	4.64	6.17
Monks-1	25.18	3.72	25.19	6.35	17.48	8.4
Monks-2	34.89	8.79	34.91	4.86	34.93	8.83
Monks-3	3.87	3.09	4.63	2.99	3.86	4.02
Pima	25.05	4.36	22.07	4.84	25.07	6.43
Tic Tac Toe	26.06	7.5	27.40	6.06	28.27	5.16
Vehicle	50.58	5.63	63.17	6.75	49.07	5.07

Table 11. Number of learning algorithm call with our method

Bases	Number of iterations	Number of iterations with	Number of iterations
Austra	2	3	2
Breast	3	5	5
Cleve	5	4	2
CRX	2	3	2
German	5	7	2
Heart	2	4	2
Iono	3	6	6
Iris	3	3	3
Monks-1	2	2	2
Monks-2	2	2	2
Monks-3	2	2	2
Pima	3	4	2
Tic Tac Toe	4	4	3
Vehicle	9	7	6

of classes. Experimentations with MIFS and ReliefF are also carried out on these same 30%. The 70% remainder are used for the learning stage. For that, we choose a 10-fold-cross-validation and learning algorithms are ID3, Sipina and Naïve Bayesian (NB). Tests before selection are also carried out on these same 70%. Tables 2, 3 and 4 show error rate and the associated Standard deviation (Sd) obtained before and after features selection respectively with ID3, Naïve Bayesian and Sipina by using our method. The results obtained with ID3 and BN are interesting. Except for some bases, we can see an error rate reduction and/or a stabilization of the results (Sd reduction). For Sipina, the results before and after selection are practically identical and sometimes there is an error rate degradation. For Cleve, Heart and German with Sipina, we can observe an important increase of the error rate. Tables 5, 6 and 7 indicate the number of selected features respectively with ID3, Sipina and Naive Bayesian. Our results are between those of ReliefF and those of MIFS. Tables 8, 9 and 10 allow us compare our method with ReliefF and MIFS. Results are sometimes equivalent. Our method obtain better results in most of case. Table 11 shows the number of iterations carried out by our method. The maximum number of iterations is about 9 (for Vehicle). The number of learning algorithm call in our method is then smaller than in pure wrapper methods.

5 Conclusion

In this article, we present a feature selection method based on preferences aggregation. It is a hybrid method between filter and wrapper approaches having the advantages of each approach and reducing theirs disadvantages :

- The influence of the selected features on the learning algorithm is taken into account. Thus, the selected features are different according to the used algorithm.
- The computational cost is largely lower than the computational cost of pure wrapper methods due to the use of a preordering.

Because of users can choose the short-sighted criteria set and the learning algorithm for wrapper stage, our method can be qualified “meta-method”.

Concerning the number of selected features, ours results are comparable and even better with those obtained by ReliefF and MIFS. Concerning the accuracy, we can observe an error rate reduction after selection.

We plan to improve our method according to two aspects. The discretization method used for the criteria values must be more suitable. We would like, also, that the result of the method of preferences aggregation is not a list of features subsets, but the optimal features subset.

References

1. Almuallim H., and Dietterich T. G. : Learning with many irrelevant features. In Proceedings of the Ninth National Conference on Artificial Intelligence, 547- 552, Menlo Park, CA: AAAI Press, 1991.
2. Almuallim, H., and Dietterich, T. G. : Efficient algorithms for identifying relevant features. In Proc. of the 9th Canadian Conference on Artificial Intelligence, Vancouver, BC (1992) 38-45.
3. Battiti, R. : Using mutual information for selecting features in supervised neural net learning, IEEE Trans. on Neural Networks (1994) vol. 5, 537–550.
4. Blake, C.L. and Merz, C.J. : UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science. <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/>, (1998).
5. Blum, A. L. and Langley, P. : Selection of relevant features and examples in machine learning. Artificial Intelligence, 245-271, 1997.
6. Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. : Classification and Regression trees. The Wadsworth Statistics/Probability Series, Wadsworth, Belmont, CA (1984).
7. De Mantaras, R.L. : A distance-based attribute selection measure for decision tree induction. Machine Learning, (1991) 6:81-92.
8. Hall, M. : Correlation-based feature selection of discrete and numeric class machine learning. In Proceedings of the International Conference on Machine Learning, pages 359-366, San Francisco, CA (2000). Morgan Kaufmann Publishers.
9. Hart, A. : Experience in the use of an inductive system in knowledge eng. In M. Bramer, editor, Research and Development in Expert Systems. Cambridge Univ. Press, Cambridge, MA, (1984).
10. John, G.H., Kohavi, R., Pflieger, K. : Irrelevant Features and the Subset Selection Problem, Proc. of the 11th International Conference on Machine Learning ICML94, (1994) 121-129.
11. Jun, B.H., Kim, C.S., Song, H.Y. and Kim, J. : A New Criterion in Selection and Discretization of Attributes for the Generation of Decision Trees , IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 19, n12, (1997) 1371--1375.

12. Kira, K., and Rendell, L. A. : The feature selection problem: Traditional methods and a new algorithm. In Tenth National Conference on Artificial Intelligence, (1992) 129-134. MIT Press.
13. Kira, K. and Rendell, L. : A practical approach to feature selection. In Proceedings of the Tenth International Conference on Machine Learning, Amherst, Massachusetts, (1992). Morgan Kaufmann.
14. Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. : Optimization by simulated annealing. *Science*, (1983) 220:671-680.
15. Kononenko, I. : Estimating attributes: analysis and extensions of Relief. In L. De Raedt and F. Bergadano, editors, *Machine Learning: ECML94*, 171-182. Springer Verlag, 1994.
16. Lallich, S., Rakotomalala, R. : Fast feature selection using partial correlation for multivalued attributes. *Proceedings of the 4th European Conference on Knowledge Discovery in Databases, PKDD 2000*, (2000) 221-231.
17. Langley, P., and Sage, S. : Oblivious decision trees and abstract cases. In *Working Notes of the AAAI94 Workshop on Case-Based Reasoning*, 1994. In press.
18. Liu, H. and Setiono, R. : Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of 7th IEEE Int'l Conference on Tools with Artificial Intelligence*, 1995.
19. Liu, H. and Setiono, R. : A probabilistic approach to feature selection-a filter solution. In *Proceeding of International Conference on Machine Learning*, (1996) 319-327.
20. Liu, H., Motoda H., and Dash M. : A monotonic measure for optimal feature selection. In *Proceedings of European Conference on Machine Learning*, (1998) 101-106.
21. Liu, H. and Motoda, H. : *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic, 1998.
22. Marcotorchino, J.-F. and Michaud, P. : Heuristic approach to the similarity aggregation problem. *Methods of Operations Research*, (1981) 43 : 395-404.
23. Marcotorchino, J.-F. : *Agrégation de similarités en classification automatique*, thèse de Doctorat d'Etat, Université Paris 6, (1981).
24. Michaud, P. : *Agrégation à la majorité 1 : Hommage à Condorcet*, Rapport du Centre Scientifique IBM-France, N°F-051, (1982).
25. Mingers, J. : Expert systems --- rule induction with statistical data. *Journal of the Operational Research Society*, (1987) 38:39-47.
26. Modrzejewski, M. : Feature Selection Using Rough Sets Theory. (1993) 213-226 of: *Proceedings of the European Conference on Machine Learning*. Springer.
27. Narendra, P.M. and Fukunaga, K. : A Branch and Bound algorithm for feature subset selection. *IEEE Transactions Computers*, C-26:917, September 1977.
28. Nicoloyannis, N., Terrenoire, M. and Tounissoux, D. : An optimisation model for aggregating preferences : A simulated annealing approach. *Health and System Science*, (1998) 2(1-2) :33-44.
29. Nicoloyannis, N., Terrenoire, M. and Tounissoux, D. : Pertinence d'une classification. *Revue Electronique sur l'Apprentissage par les Données*, (1999) 3(1) :39-49.
30. Quinlan, J. : *Introduction of Decision Trees*, *Machine Learning*, vol. 1, (1986) 81-106.
31. Shannon C.E.. A mathematical theory of communication. *Bell System Technical Journal*, 27:379--423,623--656, 1948.
32. Sheinvald, J., Dom, B. and Niblack, W. : A modelling approach to feature selection. In: *Proceedings of Tenth International Conference on Pattern Recognition*, (1990) 1:535--539.
33. Zhou, X. and Dillon, T.S. : A Statistical--Heuristic Feature Selection Criterion for Decision Tree Induction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1991) 13, 8:834-841.

34. Zighed, D. A., Rakotomalala, R., Rabaséda, S. : A discretization method of continuous attributes in induction graphs, in Proc. Of the 13th European Meetings on Cybernetics and System Research, (1996) 997-1002.
35. Walid Erray, "WF : Une méthode de sélection de variables combinant une méthode filtre rapide et une approche enveloppe", 11èmes Rencontres de la Société Francophone de Classification (SFC 04), Bordeaux, Septembre 2004.

Ranked Modelling with Feature Selection Based on the *CPL* Criterion Functions*

Leon Bobrowski^{1,2}

¹ Faculty of Computer Science, Bialystok Technical University

² Institute of Biocybernetics and Biomedical Engineering,
PAS, Warsaw, Poland

Abstract. Ranked transformations should preserve a priori given ranked relations (order) between some feature vectors. Designing ranked models includes feature selection tasks. Components of feature vectors which are not important for preserving the vectors order should be neglected. This way unimportant dimensions are greatly reduced in the feature space. It is particularly important in the case of “long” feature vectors, when a relatively small number of objects is represented in a high dimensional feature space. In the paper, we describe designing ranked models with the feature selection which is based on the minimization of convex and piecewise linear (*CPL*) functions.

Keywords: Ranked linear models, feature selection, convex and piecewise linear (*CPL*) criterion functions, linear separability of data sets.

1 Introduction

Special tools for data exploration are based on a variety of methods including: multivariate data analysis [1], data mining [2], pattern recognition [3], fuzzy sets [5], rough sets [6], or machine learning [7].

Data exploration goals may include trends for extraction on the basis of a known order between selected objects represented as feature vectors in a data set. For example, we could know that some objects are older (more developed, more efficient, more expensive, ...) than any object from the first set and they are younger (less developed, less efficient, less expensive, ...) than any object from the second set. This kind of a priori information about the order relation between selected pairs of objects can be the basis for ranked model designing. We assume here the ranked model is such a linear transformation, which preserves in a satisfactory manner the a priori knowledge on a line in the form of the order relations between selected pairs of feature vectors. The process of ranked model designing can be seen as trend induction from data sets which is based on a priori information about the data ordering.

* This work was partially supported by the W/II/1/2005 from the Bialystok University of Technology and by the 16/St/2005 grant from the Institute of Biocybernetics and Biomedical Engineering PAS.

The procedure of the ranked models design which is based on the minimisation of the convex and piecewise linear (*CPL*) criterion functions is described in the paper. These criterion functions are the sums of the positive and the negative *CPL* penalty functions which are defined through differences between the feature vectors constituting referencing dipoles [8]. This way, the task of the ranked model design can be linked to the problem of the linear separability of two sets in a given feature space. The enlargement of the criterion function by the feature cost functions allows one to include the feature selection into the procedure of designing ranked models [9].

2 Feature Vectors and Ranked Relations

We are taking into consideration the data set C built from m feature vectors \mathbf{x}_j with the fixed indexing j

$$C = \{\mathbf{x}_j\} \quad (j = 1, \dots, m) \tag{1}$$

The components (*features*) x_{ji} of the vector $\mathbf{x}_j = [x_{j1}, \dots, x_{jn}]^T$ are numerical results of the j -th object O_j examinations ($i = 1, \dots, n$). The feature vectors \mathbf{x}_j are often of a mixed type, because they represent different types of measurements (e.g. $x_i \in \{0, 1\}$) or ($x_i \in R$)).

Let the symbol “ \prec ” mean the ranked relation “*follows*” which may be fulfilled between selected feature vectors \mathbf{x}_j and \mathbf{x}_k :

$$\mathbf{x}_j \prec \mathbf{x}_k \Leftrightarrow \mathbf{x}_k \text{ follows } \mathbf{x}_j \tag{2}$$

The relation “ \prec ” between the feature vectors \mathbf{x}_j and \mathbf{x}_k means that the pair $\{\mathbf{x}_j, \mathbf{x}_k\}$ is *ranked*. The ranked relations between particular feature vectors \mathbf{x}_j and \mathbf{x}_k could result from additional information about the objects O_j and O_k .

Our aim is to design such a transformation of feature vectors \mathbf{x}_j on the *ranked line* $y = \mathbf{w}^T \mathbf{x}$, which preserves the relation “ \prec ” (2) as precisely as possible

$$y_j = y_j(\mathbf{w}) = \mathbf{w}^T \mathbf{x}_j \tag{3}$$

where $\mathbf{w} = [w_1, \dots, w_n]^T$ is the vector of parameters.

The relation “ \prec ” (2) is preserved on the line (3) if and only if the following implication holds:

$$(\forall(j,k)) \quad \mathbf{x}_j \prec \mathbf{x}_k \Rightarrow y_j(\mathbf{w}) < y_k(\mathbf{w}) \tag{4}$$

The procedure of the ranked line design can be based on the concept of positively and negatively oriented dipoles $\{\mathbf{x}_j, \mathbf{x}_{j'}\}$ [8].

Definition 1: The ranked pair $\{\mathbf{x}_j, \mathbf{x}_{j'}\}$ ($j < j'$) of the feature vectors \mathbf{x}_j and $\mathbf{x}_{j'}$ constitutes the *positively oriented dipole* $\{\mathbf{x}_j, \mathbf{x}_{j'}\}$ ($\forall(j, j') \in I^+$) if and only if $\mathbf{x}_j \prec \mathbf{x}_{j'}$

$$(\forall(j, j') \in I^+) \quad \mathbf{x}_j \prec \mathbf{x}_{j'} \tag{5}$$

Definition 2: The ranked pair $\{\mathbf{x}_j, \mathbf{x}_{j'}\}$ ($j < j'$) of the feature vectors \mathbf{x}_j and $\mathbf{x}_{j'}$ constitutes the *negatively oriented dipole* $\{\mathbf{x}_j, \mathbf{x}_{j'}\}$ ($\forall (j, j') \in I$), if and only if $\mathbf{x}_{j'} \prec \mathbf{x}_j$.

$$(\forall (j, j') \in I) \quad \mathbf{x}_{j'} \prec \mathbf{x}_j \tag{6}$$

Definition 3: The line $y(\mathbf{w}) = \mathbf{w}^T \mathbf{x}$ (3) is fully consistent (*ranked*) with the dipoles $\{\mathbf{x}_j, \mathbf{x}_{j'}\}$ orientations if and only if

$$\begin{aligned} (\forall (j, j') \in I^+) \quad y_j(\mathbf{w}) < y_{j'}(\mathbf{w}) \quad \text{and} \\ (\forall (j, j') \in I) \quad y_j(\mathbf{w}) > y_{j'}(\mathbf{w}) \end{aligned} \tag{7}$$

where I^+ and I are the sets of the positively and negatively oriented dipoles $\{\mathbf{x}_j, \mathbf{x}_{j'}\}$ ($j < j'$).

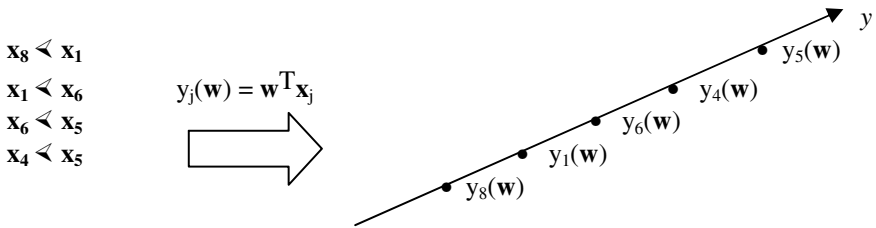


Fig. 1. An example of the order relations (2) and the ranked line (7), where $I^+ = \{(1,6), ((4,5))\}$ and $I = \{(1,8), (5,6)\}$

Let us introduce two sets C^+ and C of the differential vectors $\mathbf{r}_{jj'} = (\mathbf{x}_{j'} - \mathbf{x}_j)$ which are given by

$$\begin{aligned} C^+ &= \{\mathbf{r}_{jj'} = (\mathbf{x}_{j'} - \mathbf{x}_j): (j, j') \in I^+\} \\ C &= \{\mathbf{r}_{jj'} = (\mathbf{x}_{j'} - \mathbf{x}_j): (j, j') \in I\} \end{aligned} \tag{8}$$

We will examine the possibility of the sets separation C^+ and C by the hyperplane $H(\mathbf{w})$, which passes through the origin $\mathbf{0}$ of the feature space:

$$H(\mathbf{w}) = \{\mathbf{x}: \mathbf{w}^T \mathbf{x} = 0\} \tag{9}$$

where $\mathbf{w} = [w_1, \dots, w_n]^T$ is the vector of parameters.

Definition 4: The sets C^+ and C (8) are linearly separable with the threshold equal to zero if and only if there exists such a parameter vector \mathbf{w}^* that:

$$\begin{aligned} (\forall (j, j') \in I^+) \quad (\mathbf{w}^*)^T \mathbf{r}_{jj'} > 0 \\ (\forall (j, j') \in I) \quad (\mathbf{w}^*)^T \mathbf{r}_{jj'} < 0 \end{aligned} \tag{10}$$

The above inequalities can be represented in the following manner:

$$\begin{aligned}
 (\exists \mathbf{w}^*) (\forall (j,j') \in I^+) \quad & (\mathbf{w}^*)^T \mathbf{r}_{jj'} \geq 1 \\
 (\forall (j,j') \in I) \quad & (\mathbf{w}^*)^T \mathbf{r}_{jj'} \leq -1
 \end{aligned}
 \tag{11}$$

Remark 1: If the parameter vector \mathbf{w}^* linearly separates (11) the sets C^+ and C^- (8), then the line $y_j(\mathbf{w}^*) = (\mathbf{w}^*)^T \mathbf{x}_j$ is fully consistent (7) with the dipoles $\{\mathbf{x}_j, \mathbf{x}_{j'}\}$ orientation.

3 CPL Criterion Functions

Designing the separating hyperplane $H(\mathbf{w})$ could be carried out through the minimisation of the convex and piecewise linear (CPL) criterion function $\Phi(\mathbf{w})$ similar to the perceptron criterion function [2]. Let us introduce for this purpose the positive $\phi_{jj'}^+(\mathbf{w})$ and negative $\phi_{jj'}^-(\mathbf{w})$ penalty functions (Fig.2)

$$\begin{aligned}
 (\forall (j,j') \in I^+) \\
 \phi_{jj'}^+(\mathbf{w}) = \begin{cases} 1 - \mathbf{w}^T \mathbf{r}_{jj'} & \text{if } \mathbf{w}^T \mathbf{r}_{jj'} < 1 \\ 0 & \text{if } \mathbf{w}^T \mathbf{r}_{jj'} \geq 1 \end{cases}
 \end{aligned}
 \tag{12}$$

$$\begin{aligned}
 \text{and } (\forall (j,j') \in I) \\
 \phi_{jj'}^-(\mathbf{w}) = \begin{cases} 1 + \mathbf{w}^T \mathbf{r}_{jj'} & \text{if } \mathbf{w}^T \mathbf{r}_{jj'} > -1 \\ 0 & \text{if } \mathbf{w}^T \mathbf{r}_{jj'} \leq -1 \end{cases}
 \end{aligned}
 \tag{13}$$

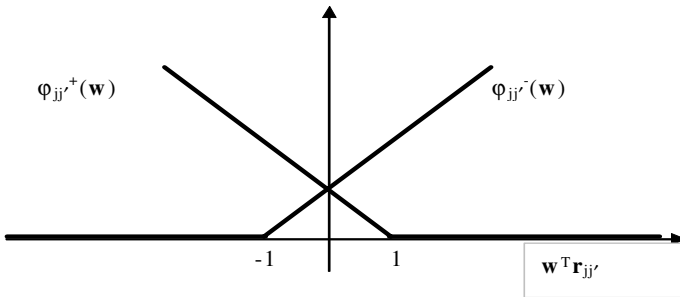


Fig. 2. The penalty functions $\phi_{jj'}^+(\mathbf{w})$ (12) and $\phi_{jj'}^-(\mathbf{w})$ (13)

The criterion function $\Phi(\mathbf{w})$ is the weighted sum of the above penalty functions

$$\Phi(\mathbf{w}) = \sum_{(j,j') \in I^+} \lambda_{jj'} \phi_{jj'}^+(\mathbf{w}) + \sum_{(j,j') \in I^-} \lambda_{jj'} \phi_{jj'}^-(\mathbf{w})
 \tag{14}$$

where $\lambda_{jj'}$ ($\lambda_{jj'} \geq 0$) is a nonnegative parameter (*price*) related to the dipole $\{\mathbf{x}_j, \mathbf{x}_{j'}\}$ ($j < j'$).

The criterion function $\Phi(\mathbf{w})$ (14) is the convex and piecewise linear (CPL) function as the sum of such type of the penalty functions $\phi_{jj'}^+(\mathbf{w})$ (12) and $\phi_{jj'}^-(\mathbf{w})$ (13). The basis exchange algorithms, similar to linear programming, allow one to find a minimum of such functions efficiently, even in the case of large, multidimensional data sets C^+ and C^- [10]:

$$\Phi^* = \Phi(\mathbf{w}^*) = \min \Phi(\mathbf{w}) \geq 0 \tag{15}$$

The optimal parameter vector \mathbf{w}^* and the minimal value Φ^* of the criterion function $\Phi(\mathbf{w})$ (11) can be applied to a variety of data ranking problems. In particular, the vector \mathbf{w}^* defining the best ranked line $y = (\mathbf{w}^*)^T \mathbf{x}$ (3) can be found this way.

Lemma 1: The minimal value Φ^* (15) of the criterion function $\Phi(\mathbf{w})$ (14) is nonnegative and equal to zero if and only if there exists such a vector \mathbf{w} that the ranking of the points $y_j(\mathbf{w})$ on the line (3) are fully consistent (Def. 3) with the relations “ \prec ” (4).

Prove: The function $\Phi(\mathbf{w})$ (14) is nonnegative as the sum of the nonnegative components $\phi_{jj'}^+(\mathbf{w})$ (12) and $\phi_{jj'}^-(\mathbf{w})$ (13). If there exists such a vector \mathbf{w}^* that the ranking of the points $y_j(\mathbf{w}^*)$ on the line (3) is fully consistent (Def. 3) with the relations “ \prec ” (4), then the sets C^+ and C^- (8) can be separated (10) by the hyperplane $H(\mathbf{w}^*)$ (9). In this case, the minimal value of the perceptron criterion function $\Phi(\mathbf{w})$ (14) is equal to zero as it results from pattern recognition theory [2]. On the other hand, if the minimal value of the criterion function $\Phi(\mathbf{w})$ (14) is equal to zero in the point \mathbf{w}^* , then the values $\phi_{jj'}^+(\mathbf{w}^*)$ and $\phi_{jj'}^-(\mathbf{w}^*)$ of all the penalty functions $1_{jj'}^+(\mathbf{w})$ (12) and $1_{jj'}^-(\mathbf{w})$ (13) have to be equal to zero. It means that the sets C^+ and C^- (8) can be separated (6) by the hyperplane $H(\mathbf{w}^*)$ (9). As the result, the ranking of the points $y_j(\mathbf{w}^*)$ on the line (3) is fully consistent (Def. 3) with the relations “ \prec ” (4). \square

Let us introduce the below hyperplanes $h_{jj'}^+$ and $h_{jj'}^-$ defined in the parameter space by the difference vectors $\mathbf{r}_{jj'} = (\mathbf{x}_{j'} - \mathbf{x}_j)$ ($j < j'$)

$$\begin{aligned} (\forall (j,j') \in I^+) \quad h_{jj'}^+ &= \{ \mathbf{w}: (\mathbf{r}_{jj'})^T \mathbf{w} = 1 \} \\ (\forall (j,j') \in I^-) \quad h_{jj'}^- &= \{ \mathbf{w}: (\mathbf{r}_{jj'})^T \mathbf{w} = -1 \} \end{aligned} \tag{16}$$

Definition 5: The parameter vector \mathbf{w} is situated on the *positive side* of the hyperplane $h_{jj'}^+$ if the inequality $(\mathbf{w})^T \mathbf{r}_{jj'} \geq 1$ is fulfilled. Similarly, the parameter vector \mathbf{w} is situated on the *positive side* of the hyperplane $h_{jj'}^-$ if the inequality $(\mathbf{w})^T \mathbf{r}_{jj'} \leq -1$ holds.

The penalty functions $1_{jj'}^+(\mathbf{w})$ (12) and $1_{jj'}^-(\mathbf{w})$ (13) are equal to zero if and only if the parameter vector \mathbf{w} is situated on the positive side of the hyperplanes $h_{jj'}^+$ and $h_{jj'}^-$ (16). The minimal value $\Phi^* = \Phi(\mathbf{w}^*)$ (15) of the criterion function $\Phi(\mathbf{w})$ (14) is equal to zero if the optimal parameter vector \mathbf{w}^* is situated on the positive side of all hyperplanes $h_{jj'}^+$ and $h_{jj'}^-$. Such a solution \mathbf{w}^* ($\Phi(\mathbf{w}^*) = 0$) exists if the sets C^+ and C^- (8) are linearly separable (10).

Remark 2: Linear independence of the vectors $\mathbf{r}_{jj'}$ constituting the sets C^+ and C^- (8) is the sufficient condition for the linear separability (10) of these sets [9].

4 Modified Criterion Function with Feature Costs

The criterion function $\Phi(\mathbf{w})$ (14) can be modified by introducing the cost function $\phi_i(\mathbf{w})$ (Fig. 3) for each feature x_i in order to search for the best feature subspace $F_1^*[m]$ [9].

$$\phi_i(\mathbf{w}) = \begin{cases} -(\mathbf{e}_i)^T \mathbf{w} & \text{if } (\mathbf{e}_i)^T \mathbf{w} < 0 \\ (\mathbf{e}_i)^T \mathbf{w} & \text{if } (\mathbf{e}_i)^T \mathbf{w} \geq 0 \end{cases} \quad (17)$$

where $\mathbf{e}_i = [0, \dots, 0, 1, 0, \dots, 0]^T$ are the unit vectors ($i=1, \dots, n$).

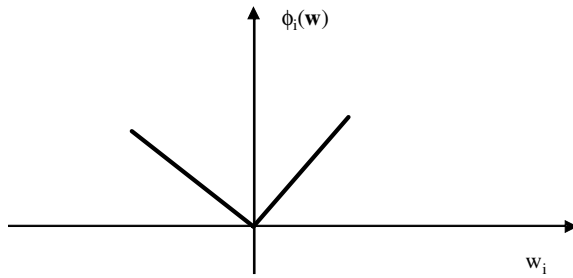


Fig. 3. The cost function $\phi_i(\mathbf{w})$ (17)

The modified criterion function $\Psi_\lambda(\mathbf{w})$ can be given in the following form [9]:

$$\Psi_\lambda(\mathbf{w}) = \Phi(\mathbf{w}) + \lambda \sum_{i \in I} \gamma_i \phi_i(\mathbf{w}) \quad (18)$$

where $\Phi(\mathbf{w})$ is given by (14), $\lambda \geq 0$, $\gamma_i > 0$, and $I = \{1, \dots, n\}$.

The function $\Psi_\lambda(\mathbf{w})$ is the sum of the perceptron criterion function $\Phi(\mathbf{w})$ (14) and the cost functions $\phi_i(\mathbf{w})$ (17) multiplied by the positive parameters γ_i . The parameters γ_i represent the costs of particular features x_i . These costs γ_i can be chosen a priori, according to our preferences.

The criterion function $\Psi_\lambda(\mathbf{w})$ (18) is the convex and piecewise linear (CPL) function as the sum of the CPL functions $\Phi(\mathbf{w})$ (14) and $\lambda \gamma_i \phi_i(\mathbf{w})$ (18). Like previously in (15), we are taking into account the point \mathbf{w}_λ^* constituting the minimal value of the criterion function $\Psi_\lambda(\mathbf{w})$:

$$\Psi_\lambda^* = \Psi_\lambda(\mathbf{w}_\lambda^*) = \min_{\mathbf{w}} \Psi_\lambda(\mathbf{w}) \quad (19)$$

The basis exchange algorithms allow one to solve efficiently also this minimisation problem [10].

The below hyperplanes h_i in the feature space can be linked to the cost functions $\phi_i(\mathbf{w})$ (17)

$$(\forall i \in I = \{1, \dots, n\}) \quad h_i = \{\mathbf{w}: (\mathbf{e}_i)^T \mathbf{w} = 0\} \tag{20}$$

The cost function $\phi_i(\mathbf{w})$ (18) is equal to zero if the point \mathbf{w} is situated on the hyperplane h_i (20).

The k -th basis $\mathbf{B}_k[n]$ of the n -dimensional feature space $F[n]$ can be constituted by any set $S_k[n]$ of n linearly independent vectors $\mathbf{r}_{jj'}$ ($(j, j') \in I^+ \cup I$) (16)) and \mathbf{e}_i ($i \in I = \{1, \dots, n\}$). The basis $\mathbf{B}_k[n]$ is the nonsingular matrix with the n rows \mathbf{b}_l constituted by the vectors $\mathbf{r}_{jj'}$ or \mathbf{e}_i :

$$\mathbf{B}_k^T[n] = [\mathbf{b}_1, \dots, \mathbf{b}_n] \tag{21}$$

where

$$\mathbf{b}_l = \mathbf{r}_{jj'} \quad \text{if} \quad \text{the vector } \mathbf{r}_{jj'} \text{ constitutes the } l\text{-th row of the matrix } \mathbf{B}_k[n] \tag{22}$$

$$\mathbf{b}_l = \mathbf{e}_i \quad \text{if} \quad \text{the vector } \mathbf{e}_i \text{ constitutes the } l\text{-th row of the matrix } \mathbf{B}_k[n]$$

The basis $\mathbf{B}_k[n]$ defines the point (the *vertex*) $\mathbf{w}_k[n]$ in the feature space in accordance with the below equation

$$\mathbf{w}_k[n] = \mathbf{B}_k^{-1}[n] \mathbf{c}_k[n] \tag{23}$$

where $\mathbf{c}_k[n] = [c_1, \dots, c_n]^T$ is the *margin* vector with the components c_l defined by the following conditions

$$c_l = 1 \quad \text{if} \quad \mathbf{r}_{jj'} ((j, j') \in I^+) \text{ (8) constitutes the } l\text{-th row of the matrix } \mathbf{B}_k[n] \tag{24}$$

$$c_l = -1 \quad \text{if} \quad \mathbf{r}_{jj'} ((j, j') \in I) \text{ (8) constitutes the } l\text{-th row of the matrix } \mathbf{B}_k[n]$$

$$c_l = 0 \quad \text{if} \quad \text{the unit vector } \mathbf{e}_i \text{ constitutes the } l\text{-th row of the matrix } \mathbf{B}_k[n]$$

It could be seen that the vertex $\mathbf{w}_k[n]$ is the point of intersection of n hyperplanes $h_{jj'}^+$ and $h_{jj'}^-$ (16) or h_i (20) in accordance with the conditions of (25).

It can be proved by applying results of linear programming theory [2], that the global minimum (19) of the criterion function $\Psi_\lambda(\mathbf{w})$ (18) can be found in one of the vertices $\mathbf{w}_k[n]$.

$$(\exists \mathbf{w}_k^*[n]) \quad (\forall \mathbf{w}) \quad \Psi_\lambda(\mathbf{w}) \geq \Psi_\lambda(\mathbf{w}_k^*[n]) \tag{25}$$

The optimal vertex $\mathbf{w}_k^*[n]$ and the related basis, the basis $\mathbf{B}_k^*[n]$, can be used in the feature selection problem.

5 Feature Selection for the Ranked Models

The optimal vertex $\mathbf{w}_k^*[n] = [w_1^*, \dots, w_n^*]^T$ (25) related to the basis $\mathbf{B}_k^*[n]$ (23) defines the ranked model (3) in the n -dimensional feature space $F[n]$.

$$y_j = (\mathbf{w}_k^*[n])^T \mathbf{x}_j[n] \tag{26}$$

Remark 3: If the unit vector \mathbf{e}_i constitutes the l -th row (22) of the optimal basis $\mathbf{B}_k^*[n]$, then the i -th feature x_i can be omitted from the feature vectors \mathbf{x}_j without the changing of the order of the points y_j on the line (26).

In order to justify the above statement let us remark, that the unit vector \mathbf{e}_i in the basis $\mathbf{B}_k^*[n]$, means that the i -th component w_i^* of the weight vector $\mathbf{w}_k^*[n]$ is equal to zero. The feature x_i related to the weight w_i^* equal to zero can be omitted without the changing of the inner products (26) value.

Remark 4: If the number m of the linearly independent vectors $\mathbf{r}_{jj'}[n] = (\mathbf{x}_{j'}[n] - \mathbf{x}_j[n])$ ($(j, j') \in I^+ \cup I$) (8) is less than the dimension n of the feature space $F[n]$ ($m < n$), then at least $n - m$ features x_i can be omitted from the vectors $\mathbf{x}_j[n]$ without changing the points y_j order on the line (26).

Neglecting the features x_i related to the unit vectors \mathbf{e}_i in the basis $\mathbf{B}_k^*[n]$ of the optimal vertex $\mathbf{w}_k^*[n]$ (26) is linked to the reduction of the feature space $F[n]$ dimension n . The reduced basis $\mathbf{B}_k^*[n']$ contains only differential vectors $\mathbf{r}_{jj'}[n'] = (\mathbf{x}_{j'}[n'] - \mathbf{x}_j[n'])$ from the feature subspace $F_1[n']$ of dimension n' .

It could be seen, that the vectors $\mathbf{r}_{jj'}[n']$ constituting the basis $\mathbf{B}_k^*[n']$ are linearly separable (11). In result, if the all vectors $\mathbf{r}_{jj'}[n']$ from the sets C^+ and C^- (8) are used in the optimal basis $\mathbf{B}_k^*[n']$, then these sets are linearly separable (10).

In the case of m linearly independent, “long” vectors $\mathbf{r}_{jj'}[n]$ ($n \gg m$) there can exist many feature subspaces $F_k[m]$ of dimension m , which assure the linear separability (11) of the sets C^+ and C^- (8) formed by the vectors $\mathbf{r}_{jj'}[n]$. The minimisation (25) of the criterion function $\Psi_\lambda(\mathbf{w})$ (18) with a small, positive values of the parameter λ ($\forall \lambda \in (0, \lambda^+)$) allows one to find the optimal feature subspace $F_1^*[m]$. It can be proved, that the minimal value $\Psi_\lambda(\mathbf{w}_k^*[m])$ (26) of the criterion function $\Psi_\lambda(\mathbf{w})$ (18) could be expressed in the below manner [11]:

$$(\forall \lambda \in [0, \lambda^+]) \quad \Psi_\lambda(\mathbf{w}_k^*[m]) = \lambda \sum_{i \in I_1^*[m]} |w_i^*| \tag{27}$$

where w_i^* are the components of the optimal, m -dimensional vertex $\mathbf{w}_k^*[m]$ (26) and $I_1^*[m]$ is the set of the indices i of such features x_i which are included in this vertex. All included features x_i have the weights w_i^* greater than zero ($(\forall i \in I_1^*[m]) w_i^* > 0$).

If the costs γ_i are equal to one, then the minimal value $\Psi_\lambda(\mathbf{w}_k^*[m])$ (27) of the function $\Psi_\lambda(\mathbf{w})$ (18) can be expressed as:

$$\Psi_\lambda^* = \Psi_\lambda(\mathbf{w}_k^*[m]) = \lambda \sum_{i \in I_1} |w_i^*| = \lambda \|\mathbf{w}_k^*[m]\|_{L_1} \tag{28}$$

where $\|\mathbf{w}_k^*[m]\|_{L_1}$ is the L_1 norm of the vector $\mathbf{w}_k^*[m]$.

In the case of such sets C^+ and C^- (8) which are linearly separable (10), the minimisation problem (19) with the function $\Psi_\lambda(\mathbf{w})$ (18) could be solved by using the following formulation [9]

$$\min_{\mathbf{w}} \{ \|\mathbf{w}\|_{L_1} : \mathbf{w} \text{ separates linearly (11) the sets } C^+ \text{ and } C^- (8) \} \tag{29}$$

The above formulation is similar to those used in the Support Vector Machines (*SVM*) method [12]. One of the important differences is such that the *SVM* method is based on the Euclidean norm $\|\mathbf{w}\|_{L_2}$, where

$$\|\mathbf{w}\|_{L_2} = (\mathbf{w}^T \mathbf{w})^{1/2} \tag{30}$$

The similarity of the expression (29) to the *SVM* approach allows one to explain in a better manner properties of the optimal vector $\mathbf{w}_k^*[m]$ which constitutes solution of the problem (29).

An efficient algorithm of the feature subspaces $F_1[m]$ exchange has been developed in order to find the optimal subspace $F_k^*[m]$ or solve the problem (29) through computations in the m -dimensional parameter spaces $F_k[m]$ instead of the initial, high dimensional feature space $F[n]$ [11].

The optimal vertex $\mathbf{w}_k^*[m]$ (25) related to the basis $\mathbf{B}_k^*[m]$ defines the ranked model $y = (\mathbf{w}_k^*[m])^T \mathbf{x}[m]$ (26) in the m -dimensional feature subspace $F_1^*[m]$. Such ranked model allows one to put new objects $\mathbf{x}[m]$ on the ranked (trend) line (3) and provides additional information concerning features x_i ($i \in I_1^*[m]$) which are the most important for preserving the discovered trend.

6 Concluding Remarks

The concept of ranked linear transformations (2) of the feature space X on the line is examined in the paper. Such lines reflect (3), to a possible extent, the relations “ \prec ” (4) between the feature vectors \mathbf{x}_j in the selected pairs $\{\mathbf{x}_j, \mathbf{x}_{j'}\}$ ($(j, j') \in I^+$) or $(j, j') \in I$). It has been shown that the ranked linear transformations (2) are linked to the concept of linear separability of some data sets.

Designing ranked linear transformations (2) can be based on minimisation of the convex and piecewise linear (*CPL*) criterion function $\Psi_\lambda(\mathbf{w})$ (18). The basis exchange algorithms, similar to linear programming, allow one to find the minimum of this function [10].

Designing ranked linear transformations allows for sequencing the feature vectors \mathbf{x}_j in a variety of manners, depending on the choice of sets I^+ and I (8) of oriented dipoles $\{\mathbf{x}_j, \mathbf{x}_{j'}\}$. Such approach allows for the experimental verification of different sequencing models. The models could be defined on the basis of the selected dipoles sets I^+ and I (8). Next, such a model could be verified on the basis of the dipoles from the testing sets and used as a tool for sequencing new feature vectors \mathbf{x}_j . The feature selection approach could indicate which features x_i are the most important in the ranked model.

The ranked linear transformations may have many applications. One of the most interesting applications could be the sequencing of genomic data or phylogenetic classification [13]. We are using a similar approach in designing tools for medical diagnosis support in the system *Hepar* [14].

References

1. Johnson R. A., Wichern D. W. *Applied Multivariate Statistical Analysis*, Prentice-Hall, Inc., Englewood Cliffs, New York, 1991
2. O. R. Duda and P. E. Hart, D. G. Stork: *Pattern Classification*, J. Wiley, New York, 2001
3. K. Fukunaga: *Statistical Pattern Recognition*, Academic Press, Inc., San Diego, 1990.
4. Fayyad U. M., Piatetsky-Shapiro G., Smyth P., Uthursamy R. (Eds.): *Advances in Knowledge Discovery and Data Mining*, AAAI Press The MIT Press, London (1996)
5. Zadeh L.A., Kacprzyk J. (Eds.), *Fuzzy Logic for the Management of Uncertainty*, John Wiley & Sons, New York, 1992
6. Pawlak Z. *Rough Sets - Theoretical Aspects of Reasoning about Data* Kluwer Academic Publishers, Dordrecht 1991
7. Michalski R. S., Bratko I., Kubat M., *Machine Learning and Data Mining*, J. Wiley, New York 1998
8. Bobrowski L.: "Piecewise-Linear Classifiers, Formal Neurons and separability of the Learning Sets", Proceedings of ICPR'96, pp. 224-228, (13th International Conference on Pattern Recognition", August 25-29, 1996, Vienna, Austria
9. Bobrowski L., Łukaszuk T., "Selection of the linearly separable feature subsets", pp. 544-549 in: *Artificial Intelligence and Soft Computing - ICAISC 2004*, Eds. L. Rutkowski et al., Springer Verlag 2004, *Springer Lecture Notes in Artificial Intelligence 3070*
10. L. Bobrowski and W.Niemiro, "A method of synthesis of linear discriminant function in the case of nonseparability". *Pattern Recognition* **17**, pp.205-210,1984
11. Bobrowski L.: *Eksploracja danych oparta na wypukłych i odcinkowo-liniowych funkcjach kryterialnych (Data mining based on convex and piecewise linear (CPL) criterion functions)* (in Polish), Technical University Białystok, 2005
12. Vapnik V. N. *Statistical Learning Theory*, J. Wiley, New York 1998
13. Bichindaritz I., Potter S., Knowledge Based Phylogenetic Classification Mining, pp. 163-172 in: *ICDM 2004*, Eds. P. Permer et al., Lipsk, Germany, Springer Verlag 2004, *Springer Lecture Notes in Artificial Intelligence 3275*
14. Bobrowski L., Wasyluk H. "Diagnosis supporting rules of the *Hepar* system", pp. 1309 – 1313 in: *MEDINFO 2001*, Proceedings of the 10-th World Congress on Medical Informatics, Ed. by V. L. Patel, R. Rogers, R Haux, IMIA, IOS Press Amsterdam 2001

A Grouping Method for Categorical Attributes Having Very Large Number of Values

Marc Boullé

France Telecom R&D, 2, Avenue Pierre Marzin,
22300 Lannion, France
marc.boullé@francetelecom.com

Abstract. In supervised machine learning, the partitioning of the values (also called grouping) of a categorical attribute aims at constructing a new synthetic attribute which keeps the information of the initial attribute and reduces the number of its values. In case of very large number of values, the risk of overfitting the data increases sharply and building good groupings becomes difficult. In this paper, we propose two new grouping methods founded on a Bayesian approach, leading to Bayes optimal groupings. The first method exploits a standard schema for grouping models and the second one extends this schema by managing a "garbage" group dedicated to the least frequent values. Extensive comparative experiments demonstrate that the new grouping methods build high quality groupings in terms of predictive quality, robustness and small number of groups.

1 Introduction

Supervised learning consists in predicting the value of a class attribute from a set of explanatory attributes. Many induction algorithms rely on discrete attributes and need to discretize continuous attributes or to group the values of categorical attributes when they are too numerous. While the discretization problem has been studied extensively in the past, the grouping problem has not been explored so deeply in the literature. However, in real data mining studies, there are many cases where the grouping of values of categorical attributes is a mandatory preprocessing step. For example, most decision trees exploit a grouping method to handle categorical attributes, in order to increase the number of instances in each node of the tree. Neural nets are based on continuous attributes and often use a 1-to-N binary encoding to preprocess categorical attributes. When the categories are too numerous, this encoding scheme might be replaced by a grouping method. This problem arises in many other classification algorithms, such as bayesian networks or logistic regression. Moreover, the grouping is a general-purpose method that is intrinsically useful in the data preparation step of the data mining process [12].

When the categorical values are both few and highly informative, grouping the values might be harmful: the optimum is to do nothing, i.e. to produce one group per value. In case of very large number of categorical values, producing good groupings

becomes harder since the risk of overfitting the data increases. In the limit situation where the number of values is the same as the number of instances, overfitting is obviously so important that efficient grouping methods should produce one single group, leading to the elimination of the attribute. Many data mining commercial packages propose to eliminate attributes having too numerous values (for example, above a threshold of 100 values). While this is reliable, potentially informative attributes might be discarded. An efficient grouping method has to compromise between information and reliability, and determine the correct number of groups.

The grouping methods can be clustered according to the search strategy of the best partition and to the grouping criterion used to evaluate the partitions. The simplest algorithm tries to find the best bipartition with one category against all the others. A more interesting approach consists in searching a bipartition of all categories. The Sequential Forward Selection method derived from [6] and evaluated by [1] is a greedy algorithm that initializes a group with the best category (against the others), and iteratively adds new categories to this first group. When the class attribute has two values, [5] have proposed in CART an optimal method to group the categories into two groups for the Gini criterion. This algorithm first sorts the categories according to the probability of the first class value, and then searches for the best split in this sorted list. In the general case of more than two class values, there is no algorithm to find the optimal grouping with K groups, apart from exhaustive search. Decision tree algorithms often manage the grouping problem with a greedy heuristic based on a bottom-up classification of the categories. The algorithm starts with single category groups and then searches for the best merge between groups. The process is reiterated until no further merge improves the grouping criterion. The CHAID algorithm [7] uses this greedy approach with a criterion close to ChiMerge [8]. The best merges are searched by minimizing the chi-square criterion applied locally to two groups: they are merged if they are statistically similar. The ID3 algorithm [13] uses the information gain criterion to evaluate categorical attributes, without any grouping. This criterion tends to favor attributes with numerous categories and [14] proposed in C4.5 to exploit the gain ratio criterion, by dividing the information gain by the entropy of the categories. The chi-square criterion has also been applied globally on the whole set of categories, with a normalized version of the chi-square value [16] such as the Cramer's V or the Tschuprow's T , in order to compare two different-size partitions.

In this paper, we present a new grouping method called MODL, which results from a similar approach as that of the MODL discretization method [3]. This method is founded on a Bayesian approach to find the most probable grouping model given the data. We first define a general family of grouping models, and second propose a prior distribution on this model space. This leads to an evaluation criterion of groupings, whose minimization conducts to the optimal grouping. We use a greedy bottom-up algorithm to optimize this criterion. Additional preprocessing and post-optimization steps are proposed in order to improve the solutions while keeping a super-linear optimization time. The MODL method comes into a standard version where the grouping model consists of a partition of the categorical values, and into an extended version where a "garbage" group is settled to incorporate the least frequent values in a preprocessing step. Extensive experiments show that the MODL method produces high quality groupings in terms of compactness, robustness and accuracy.

The remainder of the paper is organized as follows. Section 2 describes the MODL method. Section 3 proceeds with an extensive experimental evaluation.

2 The MODL Grouping Method

In this section, we present the MODL approach which results in a Bayesian evaluation criterion of groupings and the greedy heuristic used to find a near Bayes optimal grouping.

2.1 Evaluation of a Standard Grouping Model

The objective of the grouping process is to induce a set of groups from the set of values of a categorical explanatory attribute. The data sample consists of a set of instances described by pairs of values: the explanatory value and the class value. The explanatory values are categorical: they can be distinguished from each other, but they cannot *naturally* be sorted. We propose the following formal definition of a grouping model.

Definition 1: A *standard* grouping model is defined by the following properties:

1. the grouping model allows to define a partition of the categorical values into groups,
2. in each group, the distribution of the class values is defined by the frequencies of the class values in this group.

Such a grouping model is called a SGM model.

Notation:

- n : number of instances
- J : number of classes
- I : number of categorical values
- n_i : number of instances for value i
- n_{ij} : number of instances for value i and class j
- K : number of groups
- $k(i)$: index of the group containing value i
- n_k : number of instances for group k
- n_{kj} : number of instances for group k and class j

The input data can be summarized knowing n , J , I and n_i . A SGM grouping model is completely defined by the parameters $\{ K, \{k(i)\}_{1 \leq i \leq I}, \{n_{kj}\}_{1 \leq k \leq K, 1 \leq j \leq J} \}$.

In the Bayesian approach, the best model is found by maximizing the probability $P(\text{Model}/\text{Data})$ of the model given the data. Using the Bayes rule and since the probability $P(\text{Data})$ is constant under varying the model, this is equivalent to maximizing $P(\text{Model})P(\text{Data}/\text{Model})$.

Once a prior distribution of the models is fixed, the Bayesian approach allows to find the optimal model of the data, provided that the calculation of the probabilities $P(\text{Model})$ and $P(\text{Data}/\text{Model})$ is feasible. We present in Definition 2 a prior

which is essentially a uniform prior at each stage of the hierarchy of the model parameters. We also introduce a strong hypothesis of independence of the distribution of the class values. This hypothesis is often assumed (at least implicitly) by many grouping methods that try to merge similar groups and separate groups with significantly different distributions of class values. This is the case for example with the CHAID grouping method [7], which merges two adjacent groups if their distributions of class values are statistically similar (using the chi-square test of independence).

Definition 2: The following distribution prior on SGM models is called the three-stage prior:

1. the number of groups K is uniformly distributed between 1 and I ,
2. for a given number of groups K , every division of the I categorical values into K groups is equiprobable,
3. for a given group, every distribution of class values in the group is equiprobable,
4. the distributions of the class values in each group are independent from each other.

Owing to the definition of the model space and its prior distribution, the Bayes formula is applicable to exactly calculate the prior probabilities of the model and the probability of the data given the model. Theorem 1, proven in [4], introduces a Bayes optimal evaluation criterion.

Theorem 1: A SGM model distributed according to the three-stage prior is Bayes optimal for a given set of categorical values if the value of the following criterion is minimal on the set of all SGM models:

$$\log(I) + \log(B(I, K)) + \sum_{k=1}^K \log(C_{n_k+J-1}^{J-1}) + \sum_{k=1}^K \log(n_k! / n_{k,1}! n_{k,2}! \dots n_{k,J}!). \tag{1}$$

$B(I, K)$ is the number of divisions of the I values into K groups (with eventually empty groups). When $K = I$, $B(I, K)$ is the Bell number. In the general case, $B(I, K)$ can be written as a sum of Stirling numbers of the second kind:

$$B(I, K) = \sum_{k=1}^K S(I, k). \tag{2}$$

The first term of the criterion in Equation 1 stands for the choice of the number of groups, the second term for the choice of the division of the values into groups and the third term for the choice of the class distribution in each group. The last term encodes the probability of the data given the model.

2.2 Optimization of a Standard Grouping Model

Once the optimality of an evaluation criterion is established, the problem is to design a search algorithm in order to find a grouping that minimizes the criterion. In this section, we present a standard greedy bottom-up heuristic. The method starts with initial single value groups and then searches for the best merge between groups. This merge is completed if it reduces the MODL evaluation criterion of the grouping and the process is reiterated until no further merge decreases the criterion.

With a straightforward implementation of the algorithm, the method runs in $O(n^3)$ time (more precisely $O(n+I^3)$). However, the method can be optimized in $O(n^2 \cdot \log(n))$ time owing to an algorithm similar to that presented in [2]. The algorithm exploits the additivity of the evaluation criterion. Once a grouping is evaluated, the value of a new grouping resulting from the merge between two adjacent groups can be evaluated in a single step, without scanning all the other groups. Minimizing the value of the groupings after the merges is the same as maximizing the related variation of value Δ value. These Δ values can be kept in memory and sorted in a maintained sorted list (such as an AVL binary search tree for example). After a merge is completed, the Δ values need to be updated only for the new group and its adjacent groups to prepare the next merge step.

Optimized greedy bottom-up merge algorithm:

- Initialization
 - Create an elementary group for each value: $O(n)$
 - Compute the value of this initial grouping: $O(n)$
 - Compute the Δ values related to all the possible merges: $O(n^2)$
 - Sort the possible merges: $O(n^2 \cdot \log(n))$
- Optimization of the grouping

Repeat the following steps: at most n steps

 - Search for the best possible merge: $O(1)$
 - Merge and continue if the best merge decreases the grouping value
 - Compute the Δ values of the remaining group merges adjacent to the best merge: $O(n)$
 - Update the sorted list of merges: $O(n \cdot \log(n))$

In the general case, the computational complexity is not compatible with large real databases, when the categorical values becomes too numerous. In order to keep a super-linear time complexity, we extend the greedy search algorithm with several preprocessing steps whose purpose is to reduce the initial number of categorical values. For example, "pure" values (related to one single class) can be merged with no degradation of the quality of the grouping. A more harmful heuristic consists in merging the least frequent values until the desired number of values is attained.

We also add some post-optimization heuristics to improve the final grouping solution. For example, every move of a categorical value from one group to another is evaluated and the best moves are performed as long as they improve the evaluation criterion. These additional pre-processing and post-optimization heuristics are detailed in [4].

2.3 The Extended Grouping Model

When the number of categorical values increases, the grouping cost $B(I, K)$ in Equation 1 quickly rises and the potential group number falls down to 1. However, when the distribution of the categorical values is skewed, the most frequent values may be informative. A common practice in data preprocessing is to collect the least frequent values in a garbage group. In the extended grouping model presented in Definition 3, we generalize the standard grouping model by incorporating such a garbage group. After the preprocessing step, the remaining values are grouped using the standard model.

Definition 3: An *extended* grouping model is defined by the following properties:

1. the least frequent values are included into a special group called the *garbage* group,
2. the grouping model allows to define a partition of the remaining categorical values into groups,
3. in each group, the distribution of the class values is defined by the frequencies of the class values in this group.

Such a grouping model is called an EGM model.

Let F be the frequency threshold, such that the categorical values whose frequency is inferior to F are included in the garbage group. Let $I(F)$ be the number of remaining values (including the garbage group) once the preprocessing is performed. Although the extension increases the descriptive power of the model, we wish to trigger the extension only if necessary and to favor models close to the standard model, i.e. models with a small garbage frequency threshold. We express these prior preferences in Definition 4, using the universal prior for integers [15] for the distribution of F . Compared to the uniform prior, the universal prior for integers gives a higher probability to small integers with the smallest possible rate of decay. This provides a prior that favors models with small values of F .

The code length of the universal prior for integers is given by

$$L(n) = \log_2(c_0) + \log_2^*(n) = \log_2(c_0) + \sum_{j>1} \max(\log_2^{(j)}(n), 0), \tag{3}$$

where $\log_2^{(j)}(n)$ is the j^{th} composition of \log_2 ($\log_2^{(1)}(n) = \log_2(n)$, $\log_2^{(2)}(n) = \log_2(\log_2(n)) \dots$) and $c_0 = \sum_{n>1} 2^{-\log_2^*(n)} = 2.865\dots$

Definition 4: The following distribution prior on EGM models is called the three-stage prior with garbage group:

1. using or not using a garbage group are two equiprobable choices,
2. the garbage frequency threshold F is distributed according the universal prior for integers,
3. the last parameters of the grouping model, with $I(F)$ categorical values, are distributed according the three stage prior.

Owing to this prior definition, we derive an evaluation criterion for the general grouping model in Theorem 2.

Theorem 2: An EGM model distributed according to the three-stage prior with garbage group is Bayes optimal for a given set of categorical values if the value of the following criterion is minimal on the set of all EGM models:

$$\log(2) + 1_{[2, \infty[}(F)L(F)\log(2) + \log(B(I(F), K)) + \sum_{k=1}^K \log(C_{n_k+J-1}^{J-1}) + \sum_{k=1}^K \log(n_k! / n_{k,1}! n_{k,2}! \dots n_{k,J}!). \tag{4}$$

The first term corresponds to the choice of using or not using a garbage group. The second term encodes the prior probability of the garbage frequency threshold, using the code length of the universal prior for integers. The last terms are those of the criterion presented in Theorem 1.

We now have to extend the search algorithm in order to find the most probable EGM model. A first step is to sort the explanatory values by increasing frequencies. This allows to quickly compute all possible frequency thresholds F and their corresponding remaining number of values $I(F)$. Once this step is completed, a basic algorithm consists in performing the standard search algorithm on SGM models for any frequency threshold F . In the worst case, this involves $O(\sqrt{n})$ runs of the standard search algorithm, since the number of distinct frequencies F (taken from the actual frequencies of the attribute values) cannot exceed $O(\sqrt{n})$ (their sum is bounded by n). The algorithm complexity of the extended search algorithm is thus $O(n\sqrt{n} \log(n))$.

In practice, the encoding cost of the garbage group is a minor part in the criterion presented in theorem 2. Introducing a garbage group becomes relevant only when a small increase of the frequency threshold brings a large decrease of the number of remaining categorical values. This property allows designing an efficient heuristic to find the garbage frequency threshold. This greedy heuristic first evaluates the simplest extended grouping (without garbage group) and then evaluates the extended groupings by increasing the garbage frequency threshold F as long as the criterion improves. Extensive experiments show that the practical complexity of the algorithms falls down to $O(n \log(n))$, with no significant decay in the quality of the groupings.

3 Experiments

In our experimental study, we compare the MODL grouping method with other supervised grouping algorithms. In this section, we introduce the evaluation protocol, the alternative evaluated grouping methods and the evaluation results.

3.1 The Evaluation Protocol

In order to evaluate the intrinsic performance of the grouping methods and eliminate the bias of the choice of a specific induction algorithm, we use a protocol similar as [2], where each grouping method is considered as an elementary inductive method.

We choose not to use the accuracy criterion because it focuses only on the majority class value and cannot differentiate correct predictions made with probability 1 from correct predictions made with probability slightly greater than 0.5. Furthermore, many applications, especially in the marketing field, rely on the scoring of the instances and need to evaluate the probability of each class value. To evaluate the predictive quality of the groupings, we use the Kullback-Leibler divergence [9] which compared the distribution of the class values estimated from the train dataset with the distribution of the class values observed on the test dataset. For a given categorical value, let p_j be the probability of the j^{th} class value estimated on the train dataset (on the basis of the

group containing the categorical value), and q_j be the probability of the j^{th} class value observed on the test dataset (using directly the categorical value). The Kullback-Leibler divergence between the estimated distribution and the observed distribution is:

$$D(p \parallel q) = \sum_{j=1}^J p_j \log \frac{p_j}{q_j} . \quad (5)$$

The global evaluation of the predictive quality is computed as the mean of the Kullback-Leibler divergence on the test dataset. The q_j probabilities are estimated with the Laplace's estimator in order to deal with zero values.

The grouping problem is a bi-criteria problem that tries to compromise between the predictive quality and the number of groups. The optimal classifier is the Bayes classifier: in the case of an univariate classifier based on a single categorical attribute, the optimal grouping is to do nothing, *i.e.* to build one group per categorical value. In the context of data preparation, the objective is to keep most of the information contained in the attribute while decreasing the number of values. In the experiments, we collect both the predictive quality results using the Kullback-Leibler divergence and the number of groups.

In a first experiment, we compare the grouping methods considered as univariate classifiers. In a second experiment, we evaluate the results of the Naïve Bayes classifier using the grouping methods to preprocess the categorical attributes. In this experiment, the results are evaluated using the test accuracy and the robustness, computed as the ratio of the test accuracy by the train accuracy. We finally perform the same experiments using a Selective Naïve Bayes classifier.

We build a list of datasets having an increasing number of values per attribute on the basis of the Waveform dataset [5]. The Waveform dataset is composed of 5000 instances, 21 continuous attributes and a target attribute equidistributed on 3 classes. In order to build categorical attributes candidate for grouping, we discretize each continuous attribute in a preprocessing step with an equal-width unsupervised discretization. We obtain a collection of 10 datasets using 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 bin numbers for the equal-width algorithm. We build a second collection of "2D" datasets containing all the Cartesian products of the attributes. Each of these 6 datasets (for bin numbers 2, 4, 8, 16, 32, 64) contains 210 categorical attributes. We finally produce a third collection of "3D" datasets on the basis of the Cartesian products of three attributes. Each of these 4 datasets (for bin numbers 2, 4, 8, 16) contains 1330 categorical attributes. On the whole, we get 20 datasets having a large variety of categorical attributes, with average number of values per attribute ranging from 2 to more than 1000.

3.2 The Evaluated Methods

The grouping methods studied in the comparison are:

- MODL: the extended MODL method described in this paper (using a garbage group),
- MODLS: the standard MODL method (without garbage group),
- CHAID [7],

- Tschuprow [16],
- Khiops [2],
- NoGrouping: one group per value.

All these methods are based on a greedy bottom-up algorithm that iteratively merges the categories into groups, and automatically determines the number of groups in the final partition of the categories. The MODL methods are based on a Bayesian approach and incorporate preprocessing and post-optimization algorithms. The CHAID, Tschuprow and Khiops methods exploit the chi-square criterion in different manner. The CHAID method is the grouping method used in the CHAID decision tree classifier. It applies the chi-square criterion locally to two rows of the contingency table, and iteratively merges the values as long as they are statistically similar. The significance level is set to 0.95 in the experiments. The Tschuprow method is based on a global evaluation of the contingency table, and uses the Tschuprow's T normalization of the chi-square value to evaluate the partitions. The Khiops method also applies the chi-square criterion on the whole contingency table, but it evaluates the partition using the confidence level related to the chi-square criterion instead of the Tschuprow criterion. It unconditionally groups the least frequent values in a preprocessing step in order to improve the reliability of the confidence level associated with the chi-square criterion, by constraining every cell of the contingency table to have an expected value of at least 5. Furthermore, the Khiops method provides a guaranteed resistance to noise: any categorical attribute independent from the class attribute is grouped in a single terminal group with a user defined probability. This probability is set to 0.95 in the experiments.

3.3 The Univariate Experiment

The goal of the univariate experiment is to evaluate the intrinsic performance of the grouping methods, without the bias of the choice of a specific induction algorithm. The grouping are performed on each attribute of the 20 synthetic datasets derived from the Waveform dataset, using a stratified tenfold cross-validation.

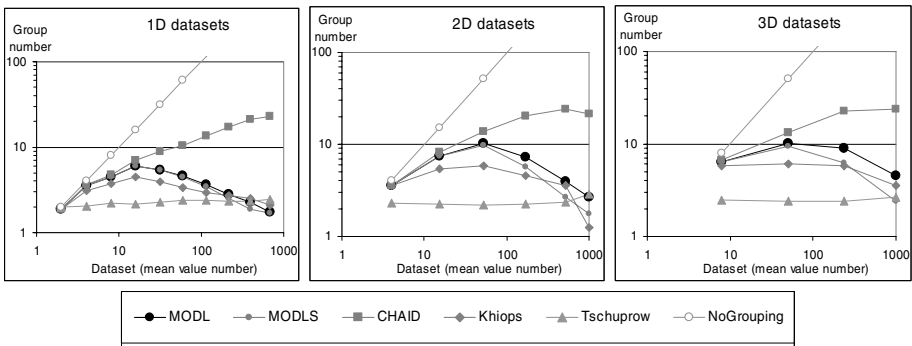


Fig. 1. Mean of the group number per attribute on the 20 datasets

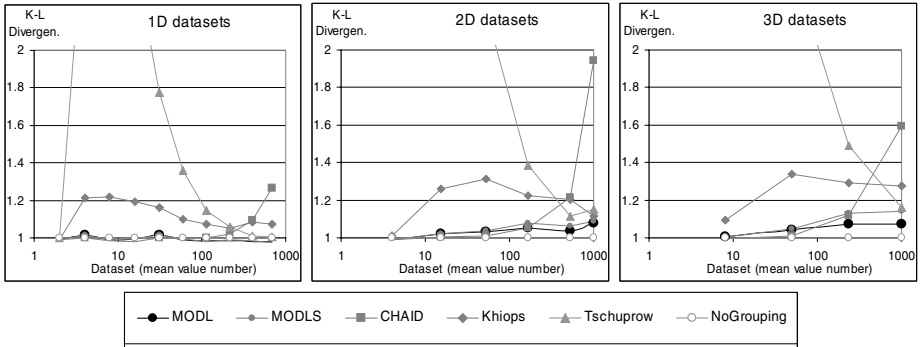


Fig. 2. Mean of the normalized Kullback-Leibler divergence per attribute on the 20 datasets

During the experiments, we collect the group number and the Kulback-Leibler divergence between the class distribution estimated on train datasets and the class distribution observed on test datasets. For each grouping method, this represents 210 measures for every 1D dataset, 2100 measures for every 2D dataset and 13300 for every 3D dataset. These results are summarized across the attributes of each dataset owing to means, in order to provide a gross estimation of the relative performances of the methods. We report the mean of the group number and of the Kullback-Leibler divergence for each dataset in Figures 1 and 2. The dataset result points are ordered by increasing bin number (from 2 bins to 1024 bins for the 1D datasets, from 2 bins to 64 bins the 2D datasets and from 2 bins to 16 bins for the 3D datasets). The result points are scaled on the x-coordinate according to the mean value number per attribute in each dataset, in order to visualize the relation between the number of values and the evaluated criterion. For the Kullback-Leibler divergence, we normalize each result by that of the NoGrouping method.

As expected, the NoGrouping method obtains the best results in term of predictive quality, at the expense of the worst number of groups. The Tschuprow method is heavily biased in favor of number of groups equal to the number of class values: it always produces between 2 and 3 groups, and obtains a very poor estimation of the class distribution (evaluated by the Kullback-Leibler divergence) as shown in Figure 2. The Khiops method suffers from its minimum frequency constraint. It produces few groups and gets a reliable estimation of the class distribution across all the datasets, whatever their mean value number per attribute. However, it fails to obtain the best groupings on most of the datasets. The CHAID and MODL methods almost reach the predictive quality of the NoGrouping method with much smaller number of groups when the mean value number is less than 100. The CHAID method produces an increasing number of groups when the number of values rises. When the number of values is very large (between 100 and 1000), it overfits the data with too many groups, and its estimation of the class distribution worsen sharply as shown in Figure 2. The MODL methods always get the best estimation of the class distribution, very close to that of the NoGrouping method. They produce an increasing number of groups when the number of values is below a few tenths and then slowly decrease the number of

groups. There is only a slight difference between the standard and the extended versions of the MODL method. When the number of values becomes very large, the extended version produces some extra groups owing to its garbage group and better approximates the class distribution.

To summarize, the MODL methods manage to get the lowest number of group without discarding the predictive quality.

3.4 The Naïve Bayes Experiment

The aim of the naïve Bayes experiment is to evaluate the impact of grouping methods on the Naïve Bayes classifier. The Naïve Bayes classifier [10] assigns the most probable class value given the explanatory attributes values, assuming independence between the attributes for each class value. The probabilities for categorical attributes are estimated using the Laplace's estimator directly on the categorical values. The results are presented in Figure 3 for the test accuracy and in Figure 4 for the robustness (evaluated as the ratio of the test accuracy by the train accuracy).

Most methods do not perform better than the NoGrouping method. This probably explains why the Naïve Bayes classifiers do not make use of groupings in the literature. The Tschuprow method is hampered by its poor estimation of the class distribution and obtains test accuracy results that are always dominated by the NoGrouping method. The Khiops method obtains good accuracy and robustness results when the number of values is below 100. For higher numbers of values, it suffers from its minimum frequency constraint and its accuracy results dramatically fall down to the accuracy of the majority classifier (33% in the Waveform dataset). The CHAID method obtains results very close to the NoGrouping method, both on the accuracy and robustness criteria. The MODL methods clearly dominate all the other methods when the two criteria are considered. On the accuracy criterion, they obtain almost the same results than the CHAID and NoGrouping methods. On the robustness criterion, they strongly dominate these two methods. Once again, there is only a minor advantage for the extended version of the MODL method compared to its standard version.

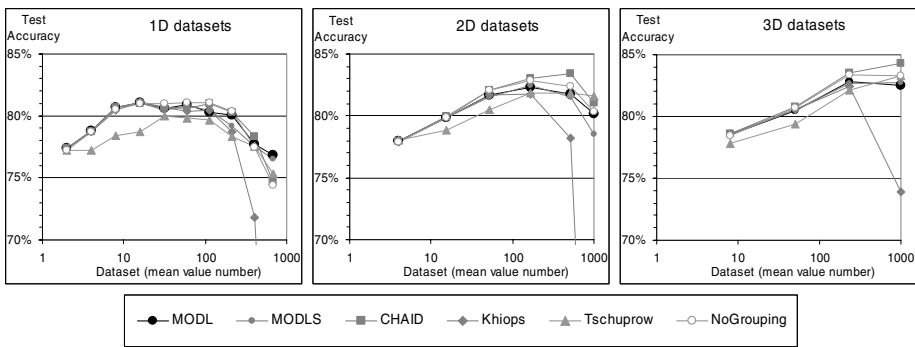


Fig. 3. Mean of the Naïve Bayes test accuracy on the 20 datasets

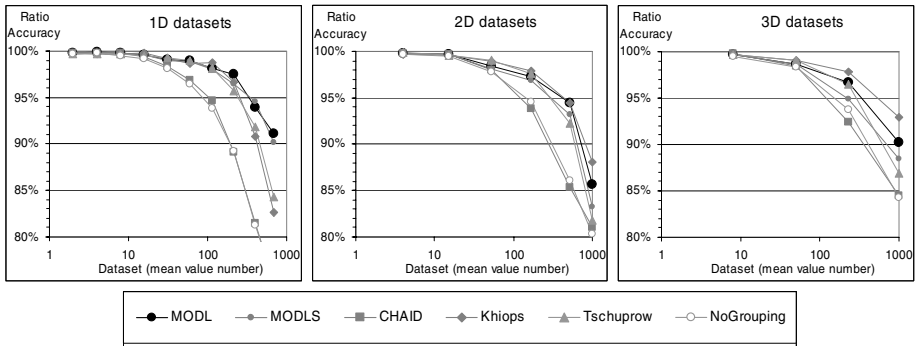


Fig. 4. Mean of the Naïve Bayes robustness on the 20 datasets

It is interesting to notice that the naïve Bayes classifier is very robust and manages to produce accurate predictions even in case of attributes having very large numbers of values. Another attractive aspect learnt from this experiment is the overall gain in test accuracy when the pairs (2D datasets) and triples (3D datasets) of attributes are considered. Using Cartesian products allows to investigate simple interactions between attributes and to go beyond the limiting independence assumption of the Naïve Bayes classifier. Although this degrades the robustness (because of a decrease in the frequency of the categorical values), this enhances the test accuracy.

3.5 The Selective Naïve Bayes Experiment

The selective naïve Bayes classifier [11] incorporates feature selection in the naïve Bayes algorithm, using a stepwise forward selection. It iteratively selects the attributes as long as there is no decay in the accuracy. We use a variant of the evaluation and stopping criterion: the area under the lift curve instead of the accuracy. The lift curve summarizes the cumulative percent of targets recovered in the top quantiles of the sample [17]. The lift curve based criterion allows a more subtle evaluation of the conditional class density than the accuracy criterion, which focuses only on the majority class.

Compared to the naïve Bayes (NB) classifier, the selective naïve Bayes (SNB) classifier is able to remove independent or redundant attributes owing to its selection process. However, it is more likely to overfit the data and requires a better evaluation of the predictive influence of each attribute. The purpose of the SNB experiment is to evaluate the impact of grouping on a classifier using an attribute selection process. The results are presented in Figure 5 for the test accuracy. The robustness results, not presented here, are very similar to those of the naïve Bayes experiment.

The Tschuprow and Khiops grouping methods suffer from their respective limitations (strong bias and minimum frequency constraint): they are constantly dominated by the other methods. The MODL, CHAID and NoGrouping achieve comparable accuracy results when the mean value number is below 100. Above this threshold, the accuracy results decrease as the mean value number still increases. The CHAID method exhibits the worst rate of decrease, followed by the NoGrouping and

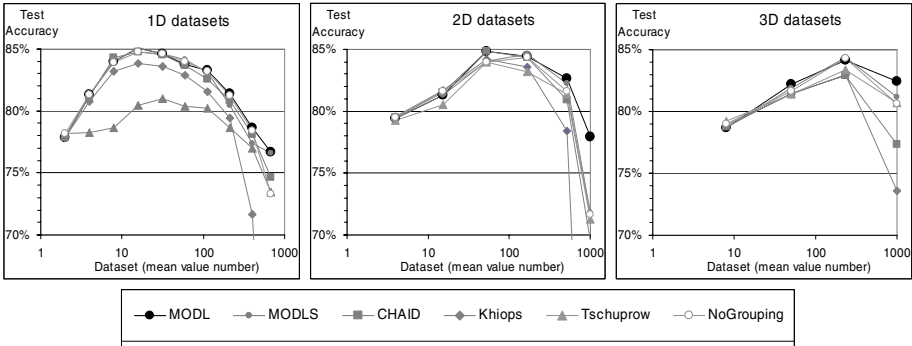


Fig. 5. Mean of the Selective Naïve Bayes test accuracy on the 20 datasets

finally the MODL methods. The extended MODL method always gets the best results. However, the benefit of the extended MODL method over the standard MODL method is still insignificant, except in the extreme case where the mean value number is close to 1000. For example, in the dataset (2D, 64 bins), the extended MODL method obtains a 77% test accuracy, about 6% above that of the standard MODL and NoGrouping methods and 8% above the CHAID method.

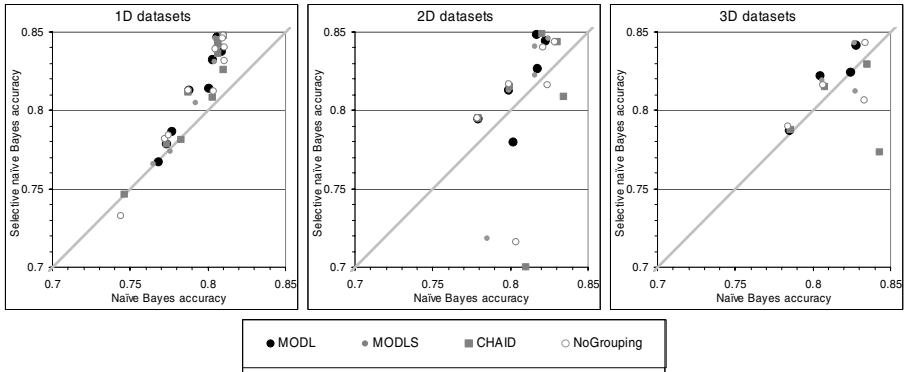


Fig. 6. Naïve Bayes versus Selective Naïve Bayes test accuracy on the 20 datasets

Apart from the grouping analysis, it is interesting to compare the results of the naïve Bayes and selective Bayes classifiers. Figure 6 reports the NB test accuracy per dataset on the x-coordinate and the SNB test accuracy per dataset on the y-coordinate for the most accurate grouping methods. Whereas the NB classifier obtains better accuracy results when pairs or triples of attributes are considered, this not the case for the SNB classifier. The SNB classifier applies its selection process to a larger set of attributes. This increases the risk of overfitting the data, so that the SNB classifier is not able to benefit from the additional information brought by the Cartesian products of attributes. On the opposite, for a given set of attributes, the SNB classifier almost

always achieves better accuracy results than the NB classifier, especially with the extended MODL algorithm. Using this grouping method, the SNB classifier improves the NB classifier accuracy results on all the 20 datasets except one (2D, 64 bins). On a whole, the extended MODL method achieves the best results with the smallest variance across the datasets.

4 Conclusion

The MODL grouping methods exploits a precise definition of a family of grouping models with a general prior. This provides a new evaluation criterion which is minimal for the Bayes optimal grouping, *i.e.* the most probable grouping given the data sample. Compared to the standard version of MODL method, the extended version incorporates a garbage group dedicated to the least frequent values.

Extensive evaluations have been performed on a collection of datasets composed of varying numbers of attributes and mean numbers of values per attribute. The most difficult dataset consists of about 5000 instances and 1000 categorical attributes, each one having 1000 values. The experiments demonstrate that the MODL methods are very efficient: they build groupings that are both robust and accurate. Compared to the CHAID method, they reduce the number of groups by up to one order of magnitude and improve the estimation of the conditional class density. They allow classifiers to take benefit of informative attributes even when their numbers of values are very large, especially with the extended version of the MODL method.

References

1. Berckman, N.C.: Value grouping for binary decision trees. Technical Report, Computer Science Department – University of Massachusetts (1995)
2. Boullé, M.: A robust method for partitioning the values of categorical attributes. *Revue des Nouvelles Technologies de l'Information, Extraction et gestion des connaissances (EGC'2004)*, RNTI-E-2, volume II, (2004a) 173-182
3. Boullé, M.: A Bayesian Approach for Supervised Discretization, *Data Mining V*, Eds Zanasi, Ebecken, Brebbia, WIT Press, (2004b) 199-208
4. Boullé, M.: MODL: une méthode quasi-optimale de groupage des valeurs d'un attribut symbolique. *Note Technique NT/FT/R&D/8611*. France Telecom R&D (2004c)
5. Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J.: *Classification and Regression Trees*. California: Wadsworth International (1984)
6. Cestnik, B., Kononenko, I. & Bratko, I.: ASSISTANT 86: A knowledge-elicitation tool for sophisticated users. In Bratko & Lavrac (Eds.), *Progress in Machine Learning*. Wilmslow, UK: Sigma Press, (1987)
7. Kass, G.V.: An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29(2) (1980) 119-127
8. Kerber, R.: Chimerge discretization of numeric attributes. *Proceedings of the 10th International Conference on Artificial Intelligence* (1991) 123-128
9. Kullback, S.: *Information Theory and Statistics*. New York: Wiley, (1959); republished by Dover, (1968)

10. Langley, P., Iba, W., & Thompson, K.: An analysis of bayesian classifiers. In Proceedings of the 10th national conference on Artificial Intelligence, AAAI Press, (1992) 223-228
11. Langley, P., & Sage, S.: Induction of Selective Bayesian Classifiers. In Proc. of the 10th Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann (1994) 399-406
12. Pyle, D.: Data Preparation for Data Mining. Morgan Kaufmann (1999)
13. Quinlan, J.R.: Induction of decision trees. *Machine Learning*, 1, (1986) 81-106
14. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
15. Rissanen, J.: A universal prior for integers and estimation by minimum description length. *Ann. Statis.* 11 (1983) 416-431
16. Ritschard, G., Zighed, D.A. & Nicoloyannis, N.: Maximisation de l'association par regroupement de lignes ou de colonnes d'un tableau croisé. *Math. & Sci. Hum.*, n°154-155, (2001) 81-98
17. Witten, I.H. & Franck, E.: Data Mining. Morgan Kaufmann (2000)

Unsupervised Learning of Visual Feature Hierarchies

Fabien Scalzo and Justus Piater*

Montefiore Institute, University of Liège,
4000 Liège, Belgium

{fscalzo, Justus.Piater}@ulg.ac.be

Abstract. We propose an unsupervised, probabilistic method for learning visual feature hierarchies. Starting from local, low-level features computed at interest point locations, the method combines these primitives into high-level abstractions. Our appearance-based learning method uses local statistical analysis between features and Expectation-Maximization to identify and code spatial correlations. Spatial correlation is asserted when two features tend to occur at the same relative position of each other. This learning scheme results in a graphical model that constitutes a probabilistic representation of a flexible visual feature hierarchy. For feature detection, evidence is propagated using Belief Propagation. Each message is represented by a Gaussian mixture where each component represents a possible location of the feature. In experiments, the proposed approach demonstrates efficient learning and robust detection of object models in the presence of clutter and occlusion and under view point changes.

1 Introduction

The visual feature representation is one of the most important issues for learning and recognition applications in computer vision. In the present work, we propose a new approach to representing and learning visual feature hierarchies in an unsupervised manner. Our hierarchical representation is inspired by the compositional nature of objects. Most objects encountered in the world, which can be either man-made or natural objects, are composed of a number of distinct constituent parts (e.g., a face contains a nose and two eyes, a phone possesses a keypad). If we examine these parts, it becomes obvious that they are in turn recursively composed of other subparts (e.g., an eye contains an iris and eyelashes, a keypad is composed of buttons). This ubiquitous observation constitutes our main motivation for arguing that a hierarchical representation must be taken into account to model objects in more flexible and realistic way.

Our long-term goal is thus to learn visual feature hierarchies that correspond to object/part hierarchies. The development of a hierarchical and probabilistic framework that is tractable in terms of complexity is a central problem for many computer vision applications such as visual tracking, object recognition and categorization, face recognition, stereo matching and image retrieval.

* This work was supported by the Wallonian Ministry of Research (D.G.T.R.E.) under Contract No. 03/1/5438.

In this paper, we combine the approaches of local, appearance-based feature detection and unsupervised model learning in a new visual feature recognition scheme. The principal objective is to obtain a probabilistic framework that allows the organization of complex visual feature models. The main idea is to use a graphical model to represent the hierarchical feature structure. In this representation, which is detailed in Section 2, the nodes correspond to the visual features. The edges model both the spatial arrangement and the statistical dependence between nodes. The formulation in terms of graphical models is attractive because it provides a statistical model of the variability of shape and appearance. The shape and appearance models are specified separately by the edges and the leaf nodes of the graph, respectively.

An unsupervised feature learning method that allows the construction of a hierarchy of visual features is introduced in Section 4. The proposed framework accumulates statistical evidence from feature observations in order to find *conspicuous coincidences* of visual feature co-occurrences. The structure of the graph is iteratively built by combining correlated features into higher-level abstractions. Our learning method is best explained by first presenting the detection process, which is described in Section 3. During detection, our scheme starts by computing local, low-level features at interest point locations. These features serve to annotate the observable leaf nodes of the graph. Then, at the second step, Belief Propagation [9], a message-passing algorithm, is used to propagate the observations up the graph, thus inferring the belief associated with higher-level features that are not directly observable. The functioning and the efficacy of our method are illustrated in Section 5. Finally, Section 6 provides a discussion of related work.

2 Representation

In this section, we introduce a new part-based and probabilistic representation of visual features (Figure 1). In the proposed graphical model, nodes represent visual features and are annotated with the detection information for a given scene. The edges represent two types of information: the relative spatial arrangement between features, and their hierarchical composition. We employ the term *visual feature* in two distinct contexts:

Primitive visual features are low-level features. They are represented by a local descriptor. For this work, we used simple descriptors constructed from normalized pixel values and located at Harris interest points [4], but our system does not rely on any particular feature detector. Any other feature detector [7] can be used to detect and extract more robust information.

Compound visual features consist of flexible geometrical combinations of other sub-features (primitive or compound features).

Formally, our graph \mathcal{G} is a mathematical object made up of two sets: a vertex set \mathcal{V} , and a directed edge set $\vec{\mathcal{E}}$. For any node $s \in \mathcal{V}$, the set of parents and the set of children are respectively defined as $U(s) = \{u_i \in \mathcal{V} | (u_i, s) \in \vec{\mathcal{E}}\}$ and $C(s) = \{c_i \in \mathcal{V} | (s, c_i) \in \vec{\mathcal{E}}\}$. Information about feature types and their specific occurrences in an image will be represented in the graph by annotations of vertices and edges, as described next.

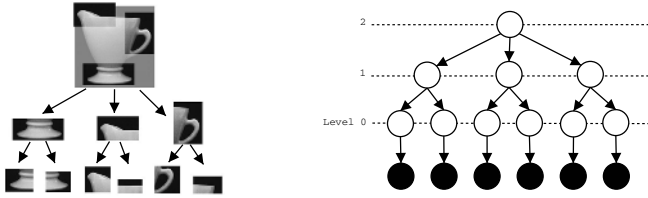


Fig. 1. Object part decomposition (left) and corresponding graphical model (right)

2.1 The Vertex Set

The vertices $s \in \mathcal{V}$ of the graph represent features. They contain the feature activations for a given image. Our graphical model associates each node with a hidden random variable $x \in \mathcal{R}^2$ representing the spatial probability distribution of the feature in the image. This random variable is continuous and defined in a two-dimensional space, where the dimensions \mathcal{X}, \mathcal{Y} are the location in the image. For simplicity, we assume that the distribution of x can be approximated by a Gaussian mixture. In order to retain information about feature orientation, we associate a mean orientation $\theta_i \in [0, 2\pi[$ to each component of the Gaussian mixture. Primitive feature classes lie at the first level of the graph. They are represented by a local descriptor and are associated with an observable random variable y , defined in the same two-dimensional space as the hidden variables x , and are likewise approximated by a Gaussian mixture.

2.2 The Edge Set

An edge $e \in \vec{\mathcal{E}}$ of the graph models two aspects of the feature structure. First, whenever an edge links two features it signifies that the destination feature is a part of the source feature. Since the observation of a part may have more or less impact on the perceived presence of a compound feature, we use a parameter $b_{x_c}^{x_u}$ to quantify the relative contribution of the subfeature x_c to the presence of the parent feature x_u .

Second, an edge describes the spatial relation between the compound feature and its subfeature in terms of the distance and the relative orientation of the subfeature with respect to the compound feature. The shape uncertainty is represented by a Gaussian probability density that models the relative position of one feature versus the other. This Gaussian, which is described by a relative location and a covariance matrix Σ , allows us to deal with deformable features and offers invariance properties.

The annotation associated with an edge, $\{x_c, x_u, d, \theta_r, \Sigma, b_{x_c}^{x_u}\}$, describes both the geometric relation (distance d , relative orientation θ_r) between two feature classes and the combination coefficient $b_{x_c}^{x_u}$. It does not contain any specific information about a given scene but represents a static view of the feature hierarchy and shape.

3 Feature Detection

In the preceding section, we defined the structure of our graphical model used to represent visual features. We now turn to the detection of a given visual feature in an image,

given its graphical-model representation. The goal of the feature detection process is to estimate the probability density function $\hat{p}(x|y)$ of features given the primitives detected in the image. The detection process can be summarized by two steps. First, we use the primitives of a scene to instantiate the observed variables y in the graph. Second, we apply Belief Propagation to infer the probability densities of the hidden variables.

3.1 Primitive Detection

Primitive detection matches the detected local descriptors in an image with the most similar existing feature descriptor at the leaf nodes of the available graphs. We use the observed descriptor parameters (location, orientation, similarity) to annotate the variable y_s of the corresponding node $s \in \mathcal{V}$. This observed variable y_s represents the degree of presence of the observed primitive feature, and is modeled by a Gaussian mixture. Each component Gaussian represents a possible location of the feature and is set to the detected descriptor location. The weight w_i of a component is inversely proportional to the Mahalanobis distance between the observed descriptor and the most similar feature class. The orientation of the feature is determined from the corresponding descriptor orientation, and is associated to each Gaussian. In summary, each component of an observed random variable y_s is defined by a location α_i , a weight w_i and an orientation θ_i that are determined by the detected descriptor occurrences.

3.2 Compound Feature Detection

The detection process that computes the presence of high-level features is based on Belief Propagation (BP) [9]. To initialize the graph, we annotate the observed variables with the detected primitives (Section 3.1). Then the main idea of this inference algorithm is to propagate information by successively transmitting upward (λ) and downward (π) messages in the graph. The λ -messages represent the possible presence of a parent feature, given the presence information of the current node. Similarly, each π -message represents the location of a node with respect to its parent nodes.

In our representation, evidence is incorporated exclusively via the variables y_i representing primitives at the leaves of the graph. Higher-level features are not directly observable. Thus, the standard BP algorithm can be simplified to a single upward propagation step where it successively estimates the probability of each node, going from the leaves to the root of the graph. This procedure is implemented by the following update rule that links each variable x of our graphical model to its child nodes $c_i \in C(x)$:

$$x = b_{c_1}^x \vartheta(c_1) + b_{c_2}^x \vartheta(c_2) + \dots + b_{c_n}^x \vartheta(c_n) \quad (1)$$

where $b_{c_i}^x$ is the combination coefficient of node c_i . The location and the orientation between a compound feature and a subfeature may be different. We therefore introduce a linear function ϑ that performs a spatial transformation. It translates the probability density of each variable c_i according to the direction and the distance of the corresponding edge $e(x, c_i)$.

J. Pearl demonstrated that BP converges to the optimal solution on tree-structured graphical models, while our graph may contain loops. However, BP has been shown empirically to provide good approximate solutions on a variety of loopy graphs.

As we explained in Section 2.1, our representation models nodes as continuous random variables approximated by mixtures of Gaussians. Likewise, the estimation of the conditional probability density of a node will take the form of a mixture of Gaussians $\hat{p}(x|y) = \sum_{i=1}^N w_{x(i)} \mathcal{G}(x; \mu_{x(i)}, \Sigma_{x(i)})$, where μ_x, Σ_x, w_x are respectively vectors of mean, covariance and weight. They are computed by combining the belief obtained at the child nodes (Eq. 2). For simplicity of notation, the following formulas are given for child nodes c_i composed of a single Gaussian (the case of mixtures is analogous).

$$\mu_x = \frac{\sum_{i=1}^{N_c} w_{c_i}^\vartheta \Sigma_{c_i}}{\sum_{i=1}^{N_c} \Sigma_{c_i}} \quad \Sigma_x = \left(\sum_{i=1}^{N_c} \frac{1}{\Sigma_{c_i}} \right)^{-1} \quad \mu_{c_i}^\vartheta = b_{c_i}^x \vartheta(c_i) \quad (2)$$

Feature Orientation. In order to estimate the most likely orientation of a feature, we use the orientations associated to each component of the current Gaussian mixture. We compute the mean orientation $\bar{\theta}_x(l)$ of mixture components weighted by their corresponding weights w_i : $\tan \bar{\theta}_x(l) = \frac{S_x(l)}{C_x(l)}$ where $C_x(l) = \sum_{i=1}^n v_i w_i \cos \theta_i$ and $S_x(l) = \sum_{i=1}^n v_i w_i \sin \theta_i$. In these equations, l is a location in the image, θ_i is the main orientation of component x_i , n is the number of components and $v_i = \mathcal{R}(l, x_i)$ is the response of Gaussian component x_i at point l .

4 Visual Feature Learning

In this section, we introduce our unsupervised feature learning method that allows the construction of a hierarchy of visual features. The general idea of our algorithm is to accumulate statistical evidence from the relative positions of observed features in order to find frequent visual feature co-occurrences. The structure of our graphical model is iteratively built by combining correlated features into new visual feature abstractions. First, the learning process votes to accumulate information on the relative position of features and it extracts the feature pairs that tend to be located in the same neighborhood. Secondly, it estimates the parameters of the geometrical relations using either Expectation-Maximization (EM) or a voting scheme. It finally creates new feature nodes in the graph by combining spatially correlated features. In the following sections, we describe the three main steps of this unsupervised learning procedure.

4.1 Spatial Correlation

The objective of this first step of our learning process is to find spatially correlated features. A spatial correlation exists between two features if they are often detected in the same neighborhood. Co-occurrence statistics are collected from multiple feature occurrences within one or across many different images. The procedure to find correlated features is summarized in Algorithm 1. After its completion, we obtain a vote array \mathcal{S} concerning the relative locations of correlated features. Before the first iteration we apply K-means clustering algorithm to the set of feature descriptors. This identifies primitive classes from the training set and is used to create the first level of the graph.

Algorithm 1 Detection of Spatial Correlations

Successively extract each image I from the training set
 Detect all features $f_I = \{f_{i_0} \dots f_{i_n}\} \in \mathcal{G}$ in image I
for each pair $[f_i, f_j]$ where f_j is observed in the neighborhood of f_i **do**
 Compute the relative position $p_r \in \mathcal{R}^2$ of f_j given f_i
 Vote for the corresponding observation $[f_i, f_j, p_r]$ in table \mathcal{S}
end for
 Keep all class pairs $[f_i, f_j]$ where $\sum_{p_r} \mathcal{S}[f_i, f_j, p_r] > t_c$

4.2 Spatial Relations

In our framework, spatial relations are defined in terms of distance and direction between features. We implemented two solutions to estimate these parameters. The first method uses the Expectation-Maximization (EM) algorithm, and the second implements a fast discrete voting scheme to find location evidence. The estimated geometrical relations are used during feature generation (Section 4.3) in order to create new features. First, however, we give some details on both methods for the estimation of spatial relations.

Expectation-Maximization. In principle, a sample of observed spatial relations x_r between two given features can be approximated by a Gaussian mixture, where each component represents a cluster of relative positions μ_k of one of the two features f_j with respect to the other, the *reference feature* f_i : $p(x_r; \Theta) = \sum_{k=1}^K w_k \mathcal{G}(x_r; \mu_k, \theta_k)$. EM is used to estimate the parameters of the spatial relation between each correlated feature pair $[f_i, f_j] \in \mathcal{S}$. It maximizes the likelihood of the observed spatial relations over the model parameters $\Theta = (w_{1\dots K}; \mu_{1\dots K}; \theta_{1\dots K})$. The Expectation (E) and Maximization (M) steps of each iteration of the algorithm are defined as follows:

Step E. Compute the current expected values of the component indicators $t_k(x_i)$, $1 \leq i \leq n$, $1 \leq k \leq K$, where n is the number of observations, K is the number of components and q is the current iteration:

$$t_k^{(q)}(x_i) = \frac{\hat{w}_k^{(q)} \mathcal{G}(x_i; \hat{\mu}_k^{(q)}, \hat{\theta}_k^{(q)})}{\sum_{l=1}^K \hat{w}_l^{(q)} \mathcal{G}(x_i; \hat{\mu}_l^{(q)}, \hat{\theta}_l^{(q)})} \quad (3)$$

Step M. Determine the value of parameters Θ^{q+1} containing the estimates $\hat{w}_k, \hat{\mu}_k, \hat{\theta}_k$ that maximize the likelihood of the data $\{x\}$ given the $t_k(x_i)$:

$$\begin{aligned} \hat{w}_k^{(q+1)} &= \frac{1}{n} \sum_{i=1}^n t_k^{(q)} & \hat{\mu}_k^{(q+1)} &= \frac{\sum_{i=1}^n t_k^{(q)}(x_i)}{\sum_{i=1}^n t_k^{(q)}} \\ \hat{\theta}_k^{(q+1)} &= \frac{\sum_{i=1}^n t_k^{(q)}(x_i - \hat{\mu}_k^{(q+1)}) (x_i - \hat{\mu}_k^{(q+1)})^T}{\sum_{i=1}^n t_k^{(q)}} \end{aligned} \quad (4)$$

In our implementation, a mixture of only two Gaussian components ($K = 2$) is used to model spatial relations. The first component represents the most probable relative position, and the second is used to model the noise. When the location μ_1 of the first

component is estimated, it is projected into a cylindrical space defined by distance d and orientation θ parameters. We store the corresponding information $[f_i, f_j, d, \theta, \Sigma]$ in a table \mathcal{T} .

Voting. A faster method to estimate spatial relations is to discretize distance and direction between features. The idea is to create a bi-dimensional histogram for every correlated feature pair $[f_i, f_j] \in \mathcal{S}$. The dimensions of these histograms are the distance d and the relative direction θ from features f_i to f_j . Each observation $[f_i, f_j, p_r]$ stored in table \mathcal{S} is projected into a cylindrical space $[d, \theta]$ and votes for the corresponding entry $[d, \theta]$ of histogram $\mathcal{H}[f_i, f_j]$. After the completion of this voting procedure, we look for significant local maxima in the 2D histograms and store them in the table \mathcal{T} . In our implementation, the distances are expressed relative to the part size and are discretized into 36 bins, while the directions are discretized into 72 bins (5-degree precision).

4.3 Feature Generation

When a reliable reciprocal spatial correlation is detected between two features $[f_i, f_j]$, the generation of a new feature in our model is straightforward. We combine these features to create a new higher-level feature by adding a new node f_n to the graph. We connect it to its subfeatures $[f_i, f_j]$ by two edges e_i, e_j that are added to $\vec{\mathcal{E}}$. Their parameters are computed using the spatial relation $\{\mu_{i,j}, \mu_{j,i}\}$ obtained from the preceding step, and are stored in table \mathcal{T} .

The generated feature is located at the midpoint between the subfeatures. Thus the distance from the subfeatures to the new feature is set to the half distance between the subfeatures $[f_i, f_j]$; $\mu_1 = \mu_{i,j}/2$, $\mu_2 = \mu_{j,i}/2$ and is copied to the new edges; $e_i(f_i, f_n) = \{\mu_1, \Sigma_1\}$, $e_j(f_j, f_n) = \{\mu_2, \Sigma_2\}$.

5 Experiments

In this section, we illustrate our visual feature learning scheme on an object recognition task using several objects of the Columbia University Object Image Library (COIL-100) [8]. This library is very commonly used in object recognition research and contains color images of 100 different objects. Each image was captured by a fixed camera at pose intervals of 5 degrees. For our experiments, we used 5 neighboring views of an object to build the corresponding object model. When the learning process is completed, the model is thus tuned for a given view of the object.

As we mentioned before, our system does not depend on any particular feature detector. We used Harris interest points and rotation invariant descriptors comprising 256 pixel values. Any other feature detector can be used to detect and extract more robust information. We deliberately used simple feature to demonstrate the functioning of our method. To estimate the primitives of each object model, we used K-Means to cluster the feature space. The number of classes was selected according to the BIC criterion [13]. For the object presented in Figure 2, the learning process used 16 feature classes (generated by K-Means) to extract correlated features of the same level in the graphical model. For the first level of the graph, it found 7 spatial relations between features that were then used to build the next level of the graph. In order to avoid excessive growth

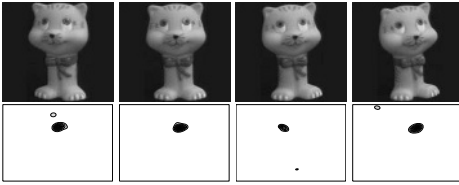


Fig. 2. Evidence map of an object model on a series of images differing in viewing angle

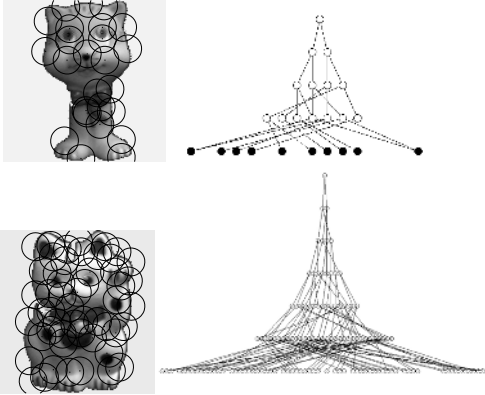


Fig. 3. Graphical models learned on two objects



Fig. 4. Cluttered scene containing three instances of the object (top) and corresponding response of the detection process for the object model (bottom). The three major peaks observed in the density map correspond to the most probable object model locations

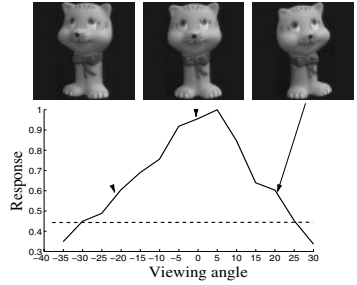


Fig. 5. Model response on a series of images differing in viewing angle by 5 degrees

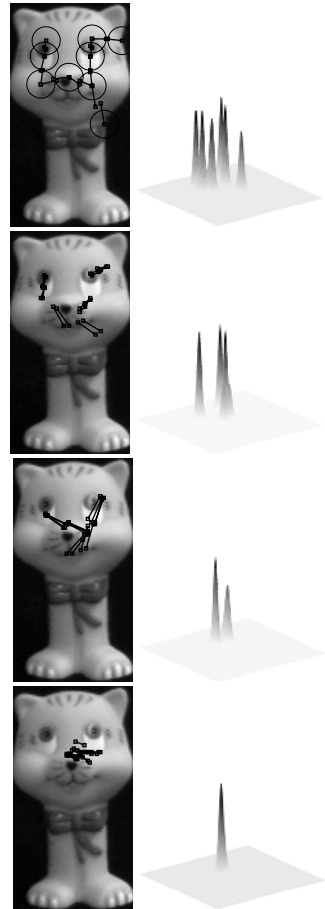


Fig. 6. Starting from the first level (top), the detection process uses the presence of primitives to infer the location of the higher level features. The sum of the density probability function for each feature of the level is shown on the right of the image

of the graph due to the feature combinatorics, we only kept the most salient spatial relations between features. Figure 3 shows the graph learned on different objects.

Figure 5 illustrates the viewpoint invariance of the object model. To generate the graph, we ran the detection process on a series of images differing in viewing angle by increments of 5 degrees. We show the maximum response of our model for each image (the detection for each image is presented in Figure 2). The model responded maximally to one of the training views, with the response gradually falling off as the image was transformed away from its preferred view. We can determine the invariance range of the object model by comparing the maximum response of all views with the responses of distractors (20 images were taken randomly from coil-100 database, some of these are presented in Figure 4). The invariance range is then defined as the range over which the model response is greater than to any of the distractor objects. For the test image presented in Figure 5, the minimum response to establish the presence of the object was approximately 0.45. We obtained an average viewpoint invariance over 50 degrees. These results are remarkable considering the fact that we did not use affine-invariant features at the leaf level.

We also tested, in Figure 4, the robustness of our model in a cluttered scene containing three instances of the object. As we explained in Section 3, the detection process starts with low-level feature detection and then propagate evidence in graph. In this image, many primitive features of the object class are detected in the image. This is due to the fact that the local appearance of some interest points is similar to the primitives of the model. However, the use of geometric relations to infer the presence of higher-level feature allows an unambiguous detection. As we can see for the object on the right of Figure 4, only a few features are needed to detect the object. The response of the model increases with the number of spatially coherent features detected. On the left, a major portion of the object is present in the image and leads to a strong response of the model.

6 Discussion

During the past years, great attention has been paid to unsupervised model learning applied to object models [10]. In these techniques, objects are represented by parts, each modeled in terms of both shape and appearance by Gaussian probability density functions. This concept, which originally operated in batch mode, has been improved by introducing incremental learning [5]. Another improvement [3] used information obtained from previously learned models. In parallel, Agarwal *et al.* [1] presented a method for learning to detect objects that is based on a sparse, part-based representations. The main limitation of these schemes lies in the representation because it only contains two levels, the features and the models.

In previous work, we used a Bayesian network classifier for constructing visual features by hierarchical composition of primitive features[11]. However, the spatial arrangement of primitives was rigid and limited the robustness of the system.

Arguing that existing techniques fail to exploit the structure of the graphical models describing many vision problems, Sudderth *et al.*[14] presented Nonparametric Belief Propagation (NBP) applicable to general distributions. Our framework can be extended using NBP instead of classical BP in order to perform a more robust detection.

A hierarchical model of recognition in the visual cortex has been proposed by Riesenhuber *et al.* [12] where spatial combinations are constructed by alternately employing a maximum and sum pooling operation. Wersing *et al.* [15] used a sparse-coding learning rule to derive local feature combinations in a visual hierarchy. However, in such models there exists no explicit representation of object structure.

In neuroscience, recent evidence [2] reinforces the idea that the coding of geometrical relations in high-level visual features is essential. Moreover, recent work suggests that the visual cortex represents objects hierarchically by parts or subfeatures [6].

The framework presented in this paper offers several significant improvements over current methods proposed in the literature. Taking advantage of graphical models, we represent shape and appearance separately. This allows us to deal with shape deformation and appearance variability. The hierarchical model presented here opens a wide door to other computer vision applications. Several directions can be pursued; the most promising and challenging is certainly the unsupervised discovery of object categories. Future work will focus on the use of Nonparametric Belief Propagation (NBP) and the integration of the hierarchical model in a supervised learning environment.

References

1. S. Agarwal and D. Roth. Learning a sparse representation for object detection. In ECCV, volume 4, pages 113–130, 2002.
2. E.E. Cooper and T.J. Wojan. Differences in the coding of spatial relations in face identification and basic-level object recognition. In *J. of Exp. Psychology*, volume 26, pages 470–488, 2000.
3. L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In ICCV, pages 1134–1141, 2003.
4. C. Harris and M. Stephens. A combined corner and edge detector. In ALVEY Vision Conference, pages 147–151, 1988.
5. S. Helmer and D. G. Lowe. Object recognition with many local features. In Workshop on Generative Model Based Vision (GMBV), Washington, D.C., July 2004.
6. Y. Lerner, T. Hendler, D. Ben-Bashat, M. Harel, and R. Malach. A hierarchical axis of object processing stages in the human visual cortex. In *Cerebral Cortex*, volume 4, pages 287–97, 2001.
7. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In CVPR, volume 2, pages 257–263, June 2003.
8. S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-100), 1996.
9. J. Pearl. Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, 1988.
10. P. Perona, R. Fergus, and A. Zisserman. Object class recognition by unsupervised scaleinvariant learning. In CVPR, volume 2, page 264, Madison, Wisconsin, June 2003.
11. J. H. Piater and R. A. Grupen. Distinctive features should be learned. In BMCV, pages 52–61, 2000.
12. M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. In *Nature Neuroscience*, volume 2, pages 1019–1025, 1999.
13. G. Schwartz. Estimating the dimension of a model. *Ann. Stat.*, 6(2):461–464, 1978.
14. E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. In CVPR, pages 605–612, 2003.
15. H. Wersing and E. Koerner. Unsupervised learning of combination features for hierarchical recognition models. In ICANN, pages 1225–1230, 2002.

Multivariate Discretization by Recursive Supervised Bipartition of Graph

Sylvain Ferrandiz^{1,2} and Marc Boullé¹

¹ France Télécom R&D, 2, avenue Pierre Marzin,
22307 Lannion Cedex, France

² Université de Caen, GREYC, Campus Côte de Nacre,
boulevard du Maréchal Juin, BP 5186
14032 Caen Cedex, France

{sylvain.ferrandiz, marc.boullé}@francetelecom.com

Abstract. In supervised learning, discretization of the continuous explanatory attributes enhances the accuracy of decision tree induction algorithms and naive Bayes classifier. Many discretization methods have been developed, leading to precise and comprehensible evaluations of the amount of information contained in one single attribute with respect to the target one.

In this paper, we discuss the multivariate notion of neighborhood, extending the univariate notion of interval. We propose an evaluation criterion of bipartitions, which is based on the Minimum Description Length (MDL) principle [1], and apply it recursively. The resulting discretization method is thus able to exploit correlations between continuous attributes. Its accuracy and robustness are evaluated on real and synthetic data sets.

1 Supervised Partitioning Problems

In supervised learning, many inductive algorithms are known to produce better models by discretizing continuous attributes. For example, the naive Bayes classifier requires the estimation of probabilities and the continuous explanatory attributes are not so easy to handle, as they often take too many different values for a direct estimation of frequencies. To circumvent this, a normal distribution of the continuous values can be assumed, but this hypothesis is not always realistic [2]. The same phenomenon leads rules extraction techniques to build poorer sets of rules. Decision tree algorithms carry out a selection process of nominal attributes and cannot handle continuous ones directly. Discretization of a continuous attribute, which consists in building intervals by merging the values of the attribute, appears to be a good solution to these problems.

Thus, as the results are easily interpretable and lead to more robust estimations of the class conditional probabilities, supervised discretization is widely use. In [2], a taxonomy of discretization methods is proposed, with three dimensions : supervised vs. unsupervised (considering a class attribute or not), global

vs. local (evaluating the partition as a whole or locally to two adjacent intervals) and static vs. dynamic (performing the discretizations in a preprocessing step or imbedding them in the inductive algorithm). This paper is placed in the supervised context.

The aim of the discretization of a single continuous explanatory attribute is to find a partition of its values which best discriminates the class distributions between groups. These groups are intervals and the evaluation of a partition is based on a compromise: fewer intervals and stronger class discrimination are better. Discrimination can be performed in many different ways. For example,

- Chimerge [3] applies chi square measure to test the independance of the distributions between groups,
- C4.5 [4] uses Shannon’s entropy based information measures to find the most informative partition,
- MDLPC [5] defines a description length measure, following the MDL principle,
- MODL [6] states a prior probability distribution, leading to a bayesian evaluation of the partitions.

The univariate case does not take into account any correlation between the explanatory attributes and fails to discover conjointly defined patterns. This fact is usually illustrated by the XOR problem (cf. Figure 1): the contributions of the axes have to be considered conjointly. Many authors have thus introduced a fourth category in the preceding taxonomy: multivariate vs. univariate (searching for cut points simultaneously or not), and proposed multivariate methods (see for examples [7] and [8]). These aim at improving rules extraction algorithms and build conjunctions of intervals. It means that considered patterns are parallelepipeds. This can be a limiting condition as underlying structures of the data are not necessarily so squared (cf. Figure 2). We then distinguish these *strongly biased* multivariate techniques from *weakly biased* multivariate ones, that consider more generic patterns. This opposition is slightly discussed in [2], where the authors talk about *feature space* and *instance space* discretizations respectively.

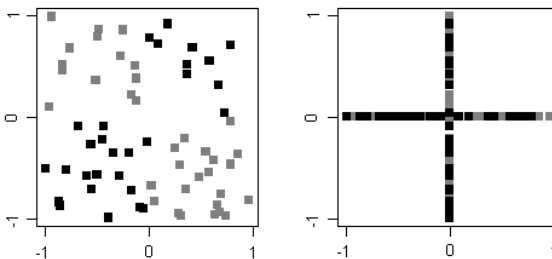


Fig. 1. The XOR problem: projection on the axes leads to an information loss

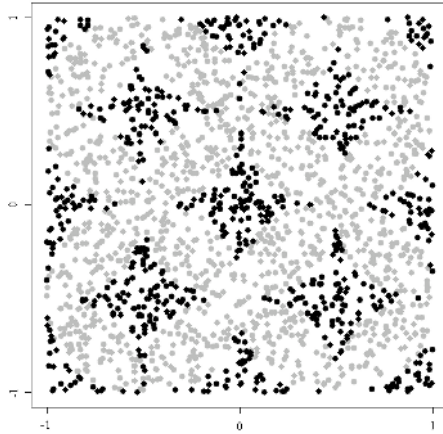


Fig. 2. A challenging synthetic dataset for strongly biased multivariate discretization methods

We present in this paper a new discretization method, which is supervised, local, static, multivariate and weakly biased. As for the MDLPC method, an evaluation criterion of a bipartition is settled following the MDL principle and applied recursively.

The remainder of the paper is organized as follow. We first set the notations (section 2). Then, we describe the MDLPC technique (section 3) and our framework (section 4). We propose a new evaluation criterion for bipartitions (section 5) and test its validity on real and synthetic datasets (section 6). Finally, we conclude and point out future works (section 7).

2 Notations

Let us set the notations we will use throughout this paper. Let $O = \{o_1, \dots, o_N\}$ be a finite set of objects. A target class l_n lying in an alphabet of size J is associated to every object o_n . For a subset A of O , $N(A)$ stands for the size of A , $J(A)$ for the number of class labels represented in A and $N_j(A)$ for the number of elements in this groups with label j ($1 \leq j \leq J$). The Shannon entropy of A , which measures the amount of information in bits needed to specify the class labels in A , is then

$$Ent(A) = - \sum_{j=1}^J \frac{N_j(A)}{N(A)} \log_2 \frac{N_j(A)}{N(A)}.$$

The problem consists in setting an evaluation criterion of the hypothesis $\mathcal{H}(A, A_1, A_2)$: split the subset A so that $A = A_1 \sqcup A_2$. We distinguish the *null* hypothesis $\mathcal{H}(A, A, \emptyset) (= \mathcal{H}(A, \emptyset, A))$ from the family of *split* hypotheses $(\mathcal{H}(A, A_1, A_2))_{A_1 \subsetneq A}$.

Following the MDL principle, a description length $l(A, A_1, A_2)$ must be assigned to each hypothesis and the best hypothesis is the one with the shortest description. Two steps are considered for the description: description of the hypothesis (leading to a description length $l_h(A, A_1, A_2)$) and description of the data given the hypothesis (leading to a description length $l_{d/h}(A, A_1, A_2)$), so that $l(A, A_1, A_2) = l_h(A, A_1, A_2) + l_{d/h}(A, A_1, A_2)$.

3 MDLPC Discretization Method

In the univariate case, O is a set of ordered real values (i.e. $o_{n_1} \leq o_{n_2}$ if $n_1 \leq n_2$) and the considered groups are intervals. The MDLPC method [5] seeks for the best split of an interval I into a couple of sub-intervals (I_1, I_2) , applying the MDL principle.

We begin by considering a split hypothesis. This is determined by the position of the boundary point, and the numbers $J(I_1), J(I_2)$ of class labels represented in I_1 and I_2 respectively. Description lengths are no other than negative log of probabilities, and assuming a uniform prior leads to write:

$$l_h(I, I_1, I_2) = \log_2(N(I) - 1) + \log_2(3^{J(I)} - 2),$$

as there is $N - 1$ possibilities for the choice of the boundary point and the number of admissible values for the couple $(J(I_1), J(I_2))$ has been evaluated to $3^{J(I)} - 2$.

The description of the data given the hypothesis consists in first specifying the frequencies of the class labels in each interval and second the exact sequences of class labels. The evaluation of the lengths is based on the entropy of the intervals I_1 and I_2 :

$$l_{d/h}(I, I_1, I_2) = J(I_1)Ent(I_1) + N(I_1)Ent(I_1) + J(I_2)Ent(I_2) + N(I_2)Ent(I_2).$$

The evaluation of $\mathcal{H}(I, I_1, I_2)$ finally relies on the following formula:

$$l(I, I_1, I_2) = \log_2(N(I) - 1) + \log_2(3^{J(I)} - 2) + J(I_1)Ent(I_1) + N(I_1)Ent(I_1) + J(I_2)Ent(I_2) + N(I_2)Ent(I_2).$$

For the null hypothesis, the class labels in I have to be described only (i.e. $l_h(I, I, \emptyset) = 0$):

$$l(I, I, \emptyset) = J(I)Ent(I) + N(I)Ent(I).$$

The MDL principle states that the best hypothesis is the one with minimal description length. As partitioning always decreases the value of the entropy function, considering the description lengths of the hypotheses allows to balance the entropy gain and eventually accept the null hypothesis. Performing recursive bipartitions with this criterion leads to a discretization of the continuous explanatory attribute at hand.

4 Multivariate Framework

Extending the univariate case mainly requires the definition of a multivariate notion of neighborhood corresponding to the notion of interval. The univariate case does not actually consider the whole set of intervals but those whose bounds are midpoints between two consecutive values. The resulting set of “patterns” is thus discrete, data dependent and induced by a simple underlying structure: the Hasse diagram, which links two consecutive elements of O .

We thus begin by supposing that a non-oriented graph structure G on O conveying a well-suited notion of proximity is provided. Some cases of natural underlying structure arise, like road networks, web graphs, etc . . . If the objects in O are tuples of an euclidean space \mathbb{R}^d and a natural structure does not exist, proximity graphs [9] provide definitions of neighborhood.

For example, as we work with vectorial data in practice, the Gabriel discrete structure can be chosen. Two multivariate instances o_1 and o_2 are *adjacent in the Gabriel sense* (cf Figure 3) if and only if

$$L(o_1, o_2)^2 \leq \min_{o \in O} L(o_1, o)^2 + L(o_2, o)^2,$$

where L is any distance measure defined on O .

The related discrete metric will be called the *Gabriel metric* on O and will be used throughout the experiments. Any prior knowledge of the user would eventually lead him to select another discrete metric, and it’s noteworthy that the use of the Gabriel one is a general choice, made without any further knowledge.

Once a discrete structure G is chosen, we define partitions on the basis of elementary “patterns” related to G . We consider the balls induced by the discrete metric δ related to G : $\delta(o_1, o_2)$ is the minimum number of edges needed to link o_1 and o_2 ($o_1, o_2 \in O$). The resulting set of balls is denoted \mathcal{B} (cf. Figure 4).

We can now express a multivariate analog of the univariate family of split hypotheses, considering balls as basic patterns. In the multivariate case, a local bipartitioning hypothesis consists in splitting a subset S of O into a ball $B \in \mathcal{B}$, included in S , and its complement. $\mathcal{H}(S, B)$ denotes such a hypothesis. As we utilize a connected discrete structure (the Gabriel graph), eventually obtaining

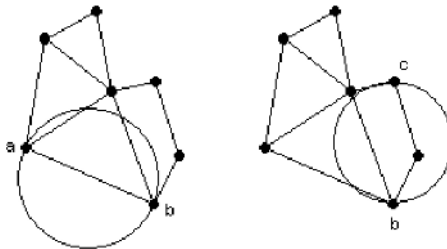


Fig. 3. Example of a Gabriel graph. The ball of diameter $[ab]$ contains no other point: a and b are Gabriel-adjacent. The ball of diameter $[bc]$ contains another point: b and c are not Gabriel-adjacent

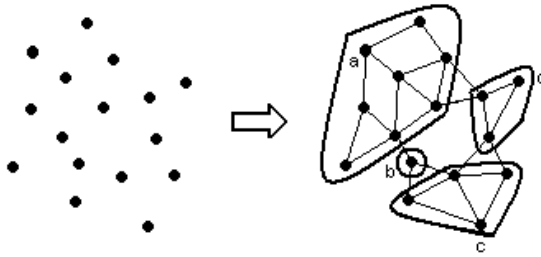


Fig. 4. Multivariate analog of intervals: examples of discrete balls. For example, the ball centered in a with radius 2 contains 8 objects of the dataset

partitions with non-connected groups can be somewhat counterintuitive. We do not try to alleviate this conceptual fact in the present paper.

5 Evaluation of a Bipartition

The proposed framework leads to the study of the hypothesis $\mathcal{H}(S, B)$, where S is a subset of O , B a ball included in S . We now introduce an evaluation criterion $l(S, B)$ for such a hypothesis. Following the MDL principle, we have to define a description length $l_h(S, B)$ of the bipartition and a description length $l_{d/h}(S, B)$ of the class labels given the bipartition.

We first consider a split hypothesis: $B \neq S$. In the univariate case, the bipartition results from the choice of a cut point. In the general case, the bipartition is determined by the ball B and the description of B relies on two parameters: its size $N(B)$ and its index in the set of balls of size $N(B)$ included in S .

Description lengths are negative log of probabilities and, if $\beta(S, B)$ stands for the number of balls of size $N(B)$ included in S , we obtain

$$l_h(S, B) = \log_2 N(S) + \log_2 \beta(S, B)$$

assuming a uniform prior.

Let us now specify the distribution of the class labels in a subset A of O (A will be S , B or $S \setminus B$). This is the same as putting the elements of A in J boxes. We begin specifying the numbers of elements to put in the j^{th} box, that is, the frequencies $(N_1(A), \dots, N_J(A))$. It then remains to give the index of the actual partition in the set of partitions of A in J groups of sizes $N_1(A), \dots, N_J(A)$.

Each possible J -uple of frequencies satisfies the property that the sum of its components equals $N(A)$. The set of possible frequencies is then of size $\binom{N(A)+J-1}{J-1}$. Counting the set of partitions of A in J groups of fixed sizes $N_1(A), \dots, N_J(A)$ is a multinomial problem and the size of this set is the multinomial coefficient $\frac{N(A)!}{N_1(A)! \dots N_J(A)!}$.

Still assuming a uniform prior, the description length of the distribution of the labels in A is then:

$$l_d(A) = \log_2 \binom{N(A) + J - 1}{J - 1} + \log_2 \frac{N(A)!}{N_1(A)! \dots N_J(A)!}.$$

For fixing $l_{d/h}(S, B)$, we suppose the distributions of the labels in B and its complement independant. This results in setting:

$$l_{d/h}(S, B) = l_d(B) + l_d(S \setminus B).$$

Finally, the description length of a split hypothesis is given by the formula:

$$\begin{aligned} l(S, B) &= \log_2 N(S) + \log_2 \beta(S, B) \\ &+ \log_2 \binom{N(B) + J - 1}{J - 1} + \log_2 \frac{N(B)!}{N_1(B)! \dots N_J(B)!} \\ &+ \log_2 \binom{N(S \setminus B) + J - 1}{J - 1} + \log_2 \frac{N(S \setminus B)!}{N_1(S \setminus B)! \dots N_J(S \setminus B)!}. \end{aligned}$$

The null hypothesis relies on the description of the size of the considered subset (S) and the distribution of the labels in S . Indicating that the size is that of S amounts to pointing the null hypothesis. Thus, $l_h(S, S, \emptyset) = \log_2 N(S)$ and $l_{d/h}(S, S, \emptyset) = l_d(S)$, giving

$$l(S, S) = \log_2 N(S) + \log_2 \binom{N(S) + J - 1}{J - 1} + \log_2 \frac{N(S)!}{N_1(S)! \dots N_J(S)!}.$$

Still, the decision results from an optimal compromise between an entropy gain and a structural cost of the considered split hypotheses, taking into account the null hypothesis as well. But the latter does not employ the Shannon entropy (as MDLPC does), replacing it by a binomial evaluation of the frequencies of the distributions. The former exploits a multinomial definition of the notion of entropy, overcoming the asymptotic validity of the Shannon entropy.

6 Experiments

The multivariate discretization algorithm consists in applying recursively the following decision rule:

1. S a subset of O (initially, $S = O$)
2. select the ball B_0 which minimizes $l(S, B)$ over the balls $B \in \mathcal{B}$ contained in S ,
3. if $l(S, B_0) < l(S, S)$, performs step 1 on $S = B$ and $S = S \setminus B$, else stop.

Constructing the Gabriel graph requires $O(N^3)$ operations. If D is the diameter of the graph, the overall number of balls is in $O(DN)$ and each decision thus results from evaluating $O(DN)$ hypotheses. Each evaluation can be performed with $O(J)$ operations, storing the $O(DN)$ sizes of the balls. At most N splits can be triggered, giving a time complexity in $O(JDN^2)$ and a space complexity in $O(DN)$ for the optimisation algorithm. In practice, the method performs

few splits and the number of available balls quickly decreases, giving an $O(N^3)$ algorithm.

We perform three experiments, one on real datasets and two on synthetic datasets. The metric is chosen to be the euclidean one. We do not consider any other metric or weighting scheme, as the experiments aim at comparing our methods with others, in a single framework.

The main advantage of partitioning methods lies in their intrinsic capacity for providing the user with an underlying structure of the analysed data. However, this structural gain may be balanced by an information loss. The first experiment aims at evaluating how our method is affected by such a flaw. We consider the resulting partition as a basic predictive model: a new instance is classified according to a majority vote in the nearest group. We thus compare the accuracy of the discretization method to the accuracy of the Nearest Neighbor rule (NN), which gives the class label of its nearest neighbor to an unseen instance [10].

The tests are performed on 11 datasets (cf Table 1) from the UCI machine learning database repository [11]. As we focus on continuous attributes, we discard the nominal attributes of the Heart, Crx and Australian database. The evaluation consists in a stratified five-fold cross-validation. The predictive accuracy of the classifiers are reported in the Table 2, as well as the robustness (i.e the ratio of the test accuracy by the train accuracy) of our classifier.

The overall predictive accuracy does not significantly suffers from the partitioning of the data (72% against 73%). But with some datasets (Iris, Wine, Vehicle), the disadvantage of making local decision is evidenced. Indeed, as illustrated by the Figure 5, a succession of local decisions can lead to the constitution of some border groups, which is especially harmful in the context of separable distributions, producing a decrease of the accuracy. While our method takes on a safe approach, handling the boundary data with cautions, the NN rule builds more hazardous decision boundaries without being penalized in term of test accuracy.

Table 1. Tested datasets

Dataset	Size	Continuous Attributes	Class Values	Majority Class
Iris	150	4	3	0.33
Wine	178	13	3	0.40
Heart	270	10	2	0.56
Bupa	345	6	2	0.58
Ionosphere	351	34	2	0.64
Crx	690	6	2	0.56
Australian	690	6	2	0.56
Breast	699	9	2	0.66
Pima	768	8	2	0.65
Vehicle	846	18	4	0.26
German	1000	24	2	0.7

Table 2. Predictive accuracy and robustness of our method and predictive accuracy of the NN rule for the tested datasets

Dataset	Test accuracy		Robustness	
	Partition	NN	Partition	NN
Iris	0.92 ± 0.05	0.96 ± 0.02	0.98 ± 0.06	0.96 ± 0.02
Wine	0.69 ± 0.09	0.76 ± 0.07	0.90 ± 0.14	0.76 ± 0.07
Heart	0.62 ± 0.04	0.55 ± 0.03	0.88 ± 0.04	0.55 ± 0.03
Bupa	0.61 ± 0.06	0.61 ± 0.05	0.85 ± 0.07	0.61 ± 0.05
Ionosphere	0.85 ± 0.04	0.87 ± 0.02	0.99 ± 0.04	0.87 ± 0.02
Crx	0.66 ± 0.03	0.64 ± 0.05	0.90 ± 0.06	0.64 ± 0.05
Australian	0.69 ± 0.02	0.68 ± 0.02	0.95 ± 0.05	0.68 ± 0.02
Breast	0.97 ± 0.01	0.96 ± 0.01	1.00 ± 0.01	0.96 ± 0.01
Pima	0.68 ± 0.01	0.68 ± 0.02	0.94 ± 0.04	0.68 ± 0.02
Vehicle	0.54 ± 0.04	0.65 ± 0.02	0.90 ± 0.07	0.65 ± 0.02
German	0.70 ± 0.01	0.67 ± 0.02	0.96 ± 0.01	0.67 ± 0.02
Mean	0.72	0.73	0.93	0.73

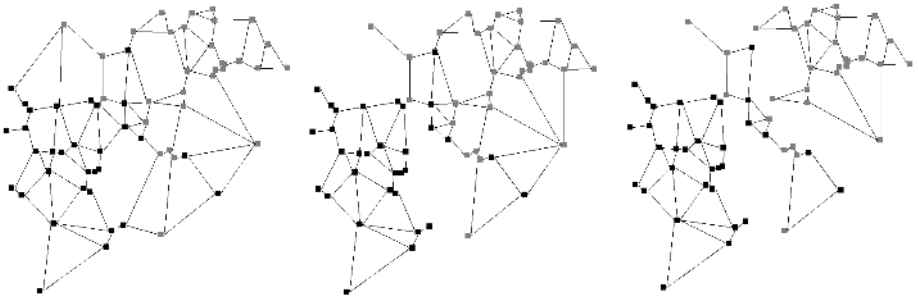


Fig. 5. Partitioning of a 2 separable classes problem: creation of a buffer zone, containing a mixture of the two classes

Table 3. Predictive accuracy, robustness and number of groups of our method, C4.5 and the NN rule on the “challenging” dataset

Method	Test accuracy	Robustness	Group number
Partition	0.83 ± 0.01	0.95 ± 0.01	29.5 ± 0.35
C4.5	0.71 ± 0.04	0.94 ± 0.01	17 ± 1.41
NN	0.90 ± 0.00	0.90 ± 0.00	-

The robustness of the NN rule is equal to its test accuracy, and we observe that building a well-suited partition of the data sharply increases the robustness of the prediction (0.93 against 0.73).

In a second experiment, we compare our method and the well-known decision tree algorithm C4.5 when faced with the challenging pattern presented in Figure 2. The dataset contains 2000 instances and we carry out a stratified two-

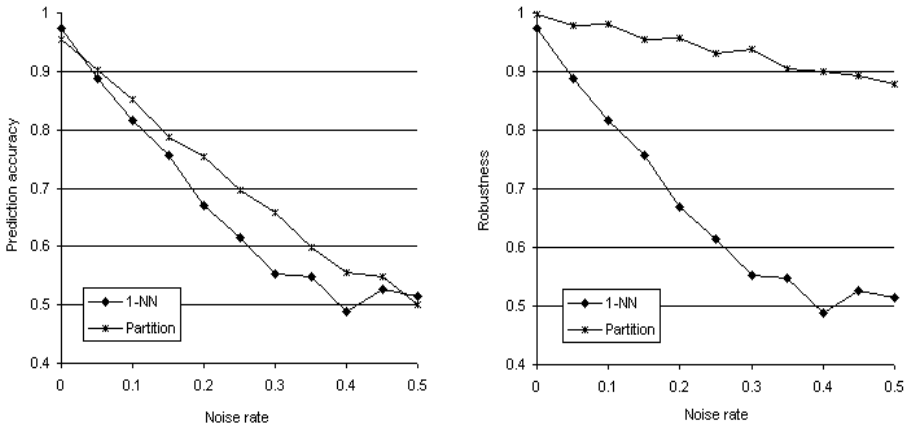


Fig. 6. Evolution of the predictive accuracy and the robustness with the mislabelled data rate of the partitioning technique and the NN rule on the XOR pattern

fold cross-validation. We report the predictive accuracy, the robustness and the number of groups in the Table 3.

From this experiment, we notice quite a strong difference between the predictive performances: our method performs a better detection than C4.5 (0.83 against 0.71). This is not surprising and illustrates the distinction between weakly and strongly biased multivariate partitioning. C4.5, which is a strongly biased method, is forced to detect parallelepipeds, limiting its detection ability as evidenced on this example. This experiment shows the robustness of the partitioning methods once again.

On the negative side, we notice a loss of predictive accuracy of our method compared with the NN rule. Examining the two produced partitions, we find that after the detection of a few clean balls (i.e objects in a ball sharing the same class label), a group containing about 600 instances marked Grey and 100 marked Black remains uncut. As the set of balls is updated by deleting balls only, the descriptive capacity of our method becomes poorer after each triggered cut. This results from the fact that we consider balls defined in the whole set O and not a locally defined set of balls. As the method makes local optimizations, performing better updates would enhance its predictive accuracy.

The third experiment consists in evaluating the tolerance of our method to the presence of mislabelled data. The method is applied to 11 Datasets, each containing 1000 instances uniformly generated in $[-1, 1] \times [-1, 1]$, representing the XOR problem with increasing mislabelled data rate, from 0 (XOR problem) to 0.5 (pure noise). The evolution of the predictive accuracy and the robustness (evaluated by a stratified 5-fold cross-validation) is shown in Figure 6, and compared with NN rule results again.

The expected optimal accuracy curve is the line passing through $(0, 1)$ and $(0.5, 0.5)$. The partitioning algorithm is up to 10% more accurate than the NN

rule and far more robust. This is its main advantage: still building accurate and robust partitions in presence of noise.

7 Conclusion and Further Works

In this paper, we have discussed the usefulness of supervised partitioning methods for data preparation in the univariate case. We have proposed an extension to the multivariate case, relying on the multivariate definition of discrete neighborhood by means of a non-oriented graph structure. A framework for supervised bipartitioning has been proposed, which applied recursively leads to a new multivariate discretization algorithm. Finally, this algorithm has been tested on real and synthetic datasets.

The proposed method builds an underlying structure of the data, producing understandable results without fitting parameters and without loss of predictive information (as shown by the experiments on real datasets). Defining basic patterns (the balls) from the data allows the technique to better partition the dataset, compared with classical strongly biased multivariate algorithm like C4.5. Furthermore, its demonstrated robustness is a main advantage, particularly since it's very tolerant to the presence of noise.

Still, more experiments have to be carried out. In the reported experiments, our method is evaluated as a classifier not as a data preparation technique. We plan to evaluate the impact of our method when considered as a preprocessing step of a naive bayes classifier, for example. Furthermore, the presented criterion can be improved, by considering local sets of balls rather than updating the global set.

Acknowledgement

We are grateful to the anonymous referees who provided helpful suggestions, and to Fabrice Clérot, Senior Expert at France Télécom R&D, for his careful proofreading of the manuscript and his comments.

References

- [1] Rissanen, J.: Modeling by shortest data description. *Automatica* **14** (1978) 465–471
- [2] Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In *Proc. of the 12th ICML* (1995) 194–202
- [3] Kerber, R.: Chimerge discretization of numeric attributes. *Tenth International Conference on Artificial Intelligence* (1991) 123–128
- [4] Quinlan, J.R.: *C4.5: Programs for machine learning*. Morgan Kaufmann (1993)
- [5] Fayyad, U., Irani, K.: On the handling of continuous-valued attributes in decision tree generation. *Machine Learning* **8** (1992) 87–102
- [6] Boullé, M.: A bayesian approach for supervised discretization. *Data Mining V*, Zanasi and Ebecken and Brebbia, WIT Press (2004) 199–208

- [7] Bay, S.: Multivariate discretization of continuous variables for set mining. In Proc. of the 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (2000) 315–319
- [8] Kwedlo, W., Kretowski, M.: An evolutionary algorithm using multivariate discretization for decision rule induction. In Proc. of the European Conference on Principles of Data Mining and Knowledge Discovery (1999) 392–397
- [9] Jaromczyk, J.W., Toussaint, G.T.: Relative neighborhood graphs and their relatives. P-IEEE **80** (1992) 1502–1517
- [10] Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. Institute of Electrical and Electronics Engineers Transactions on Information Theory **13** (1967) 21–27
- [11] Blake, C.L., Merz, C.J.: UCI repository of machine learning databases. <http://www.ics.uci.edu/~mllearn/MLRepository.html>

CorePhrase: Keyphrase Extraction for Document Clustering

Khaled M. Hammouda¹, Diego N. Matute², and Mohamed S. Kamel³

¹ Department of Systems Design Engineering

² School of Computer Science

³ Department of Electrical and Computer Engineering,

Pattern Analysis and Machine Intelligence (PAMI)

Research Group, University of Waterloo,

Waterloo ON N2L3G1, Canada

{hammouda, dnmatute, mkamel}@pami.uwaterloo.ca

Abstract. The ability to discover the topic of a large set of text documents using relevant keyphrases is usually regarded as a very tedious task if done by hand. Automatic keyphrase extraction from multi-document data sets or text clusters provides a very compact summary of the contents of the clusters, which often helps in locating information easily. We introduce an algorithm for topic discovery using keyphrase extraction from multi-document sets and clusters based on frequent and significant shared phrases between documents. The keyphrases extracted by the algorithm are highly accurate and fit the cluster topic. The algorithm is independent of the domain of the documents. Subjective as well as quantitative evaluation show that the algorithm outperforms keyword-based cluster-labeling algorithms, and is capable of accurately discovering the topic, and often ranking it in the top one or two extracted keyphrases.

1 Introduction

The abundance of information in text form has been both a blessing and a plague. There is always a need to summarize information into compact form that could be easily absorbed. The challenge is to extract the essence of text documents and present it in a compact form that identifies their topic(s). *Keyphrase extraction*, which is a text mining task, extracts highly relevant phrases from documents.

Turney [1] lists over a dozen applications that utilizes keyphrase extraction. For example, providing mini-summaries of large documents, highlighting keyphrases in text, text compression, constructing human-readable keyphrase-based indexes, interactive query refinement by suggesting improvements to queries, document clustering, and document classification.

Keyphrase extraction algorithms fall into two categories: keyphrase extraction from single documents, which is often posed as a supervised learning task; and keyphrase extraction from a set of documents, which is an unsupervised learning task that tries to *discover* the topics rather than *learn* from examples.

Extraction vs. Construction. There are two ways to finding relevant key phrases in text: either to *extract* them from existing phrases in the text, or to automatically *construct* them [2]. Construction of keyphrases is regarded as a more intelligent way of summarizing text, but in practice is more difficult.

Keyphrases vs. Keywords. A *keyphrase* is “a sequence of one or more words that is considered highly relevant”, while a *keyword* is “a single word that is highly relevant.” An arbitrary combination of keywords does not necessarily constitute a keyphrase; neither do the constituents of a keyphrase necessarily represent individual keywords.

In this paper we present a highly accurate method for extracting keyphrases from multi-document sets or clusters, with no prior knowledge about the documents. The algorithm is called **CorePhrase**, and is based on finding a set of core phrases from a document cluster.

CorePhrase works by extracting a list of candidate keyphrases by intersecting documents using a graph-based model of the phrases in the documents. This is facilitated through a powerful phrase-based document indexing model [3].

Features of the extracted candidate keyphrases are then calculated, and phrases are ranked based on their features. The top phrases are output as the descriptive topic of the document cluster. Results show that the extracted keyphrases are highly relevant to the topic of the document set. Figure 1 illustrates the different components of the keyphrase extraction system.

The work presented here assumes that: (1) keyphrases exist in the text and are not automatically generated; (2) the algorithm *discovers* keyphrases rather than *learns* how to extract them; and (3) if used with text clustering, the algorithm is

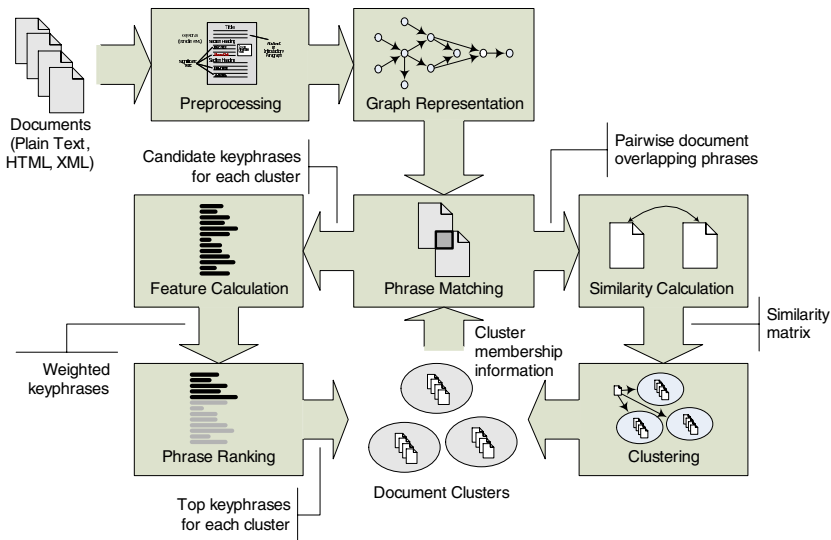


Fig. 1. CorePhrase Keyphrase Extraction System

not concerned with how the clusters are generated; it extracts keyphrases from already clustered documents.

The paper is organized as follows. Section 2 discusses related work in keyphrase extraction. The CorePhrase algorithm is presented in section 3. Experimental results are presented and discussed in section 4. Finally we discuss some conclusions and outline future work in section 5.

2 Related Work

Two popular single-document keyphrase extraction algorithms are: *Extractor* by Turney [1], and *Kea* by Frank *et al* [2]. *Extractor* uses machine learning to extract keyphrases from individual documents, which employs a genetic algorithm to tune its parameters. Evaluation is based on the number of extracted keyphrases that match human generated keyphrases, which is claimed to achieve human-like performance, but could be biased towards the trained data. *Kea* is a single document summarizer, which employs a Bayesian supervised learning approach to build a model out of training data, then applies the model to unseen documents for keyphrase extraction.

Methods using keyword-based cluster labeling include that proposed by Neto *et al* [4], which uses Autoclass-based clustering with top frequent keywords for cluster labels. In addition to keyword-based cluster summarization, it is also used to perform single document summarization, using a variation of the popular $tf \times idf$ weighting scheme to score phrases.

An information theoretic based approach for phrase extraction from multi-documents has been proposed by Bakus *et al* [5]. The method finds hierarchical combinations of statistically significant phrases.

Mani and Bloedorn suggested a method for multi-document summarization based on a graph representation based on concepts in the text [6]. Also another system for topic identification is TopCat [7]. It uses a series of natural language processing, frequent itemset analysis, and clustering steps to identify the topics in a document collection.

3 The CorePhrase Algorithm

CorePhrase works by first constructing a list of candidate keyphrases, scoring each candidate keyphrase according to some criteria, ranking the keyphrases by score, and finally selecting a number of the top ranking keyphrases for output.

3.1 Extraction of Candidate Keyphrases

Candidate keyphrases naturally lie at the *intersection* of the document cluster. The CorePhrase algorithm compares every pair of documents to extract matching phrases. This process of matching every pair of documents is inherently $O(n^2)$. However, by using a proven method of document phrase indexing

graph structure, known as the Document Index Graph (DIG), the algorithm can achieve this goal in near-linear time [3].

In essence, what the DIG model does is to keep a cumulative graph representing currently processed documents: $G_i = G_{i-1} \cup g_i$, where g_i is the subgraph representation of a new document. Upon introducing a new document, its subgraph is matched with the existing cumulative graph to extract the matching phrases between the new document and all previous documents. That is, the list of matching phrases between document \mathbf{d}_i and previous documents is given by $M_i = g_i \cap G_{i-1}$. The graph maintains complete phrase structure identifying the containing document and phrase location, so cycles can be uniquely identified. This process produces complete phrase-matching output between every pair of documents in near-linear time, with arbitrary length phrases. Figure 2 illustrates the process of phrase matching between two documents. In the figure, the two subgraphs of two documents are matched to get the list of phrases shared between them.

Since this method outputs matching phrases for each new document, it is essential to keep a *master list*, M , of unique matched phrases, which will be used as the list of candidate keyphrases. The following simple procedure keeps this list updated:

```

1: {calculate  $M_i$  for document  $\mathbf{d}_i$ }
    $M_{ij} = \{p_{ij}: 1 < j < i\}$ : matching phrases between  $\mathbf{d}_i$  and  $\mathbf{d}_j$ 
    $M_i = \{M_{ij}\}$ : matching phrases of  $\mathbf{d}_i$ 
2: for each phrase  $p_{ij}$  in  $M_i$  do
3:   if phrase  $p_{ij}$  is in master list  $M$  then
4:     add feature vector  $\mathbf{p}_i$  to  $p_{ij}$  in  $M$ 
5:     add feature vector  $\mathbf{p}_j$  to  $p_{ij}$  in  $M$  if not present
6:   else
7:     add  $p_{ij}$  to  $M$ 
8:     add feature vectors  $\mathbf{p}_i$  and  $\mathbf{p}_j$  to  $p_{ij}$  in  $M$ 
9:   end if
10: end for
11: for each unique phrase  $p_k$  in  $M$  do
12:   calculate averages of feature vectors associated with  $p_k$ 
13: end for

```

3.2 Phrase Features

Quantitative features are needed to judge the quality of the candidate keyphrases. Each candidate keyphrase p is assigned the following features:

df: **document frequency**; the number of documents in which the phrase appeared, normalized by the total number of documents.

$$df = \frac{|\text{documents containing } p|}{|\text{all documents}|}$$

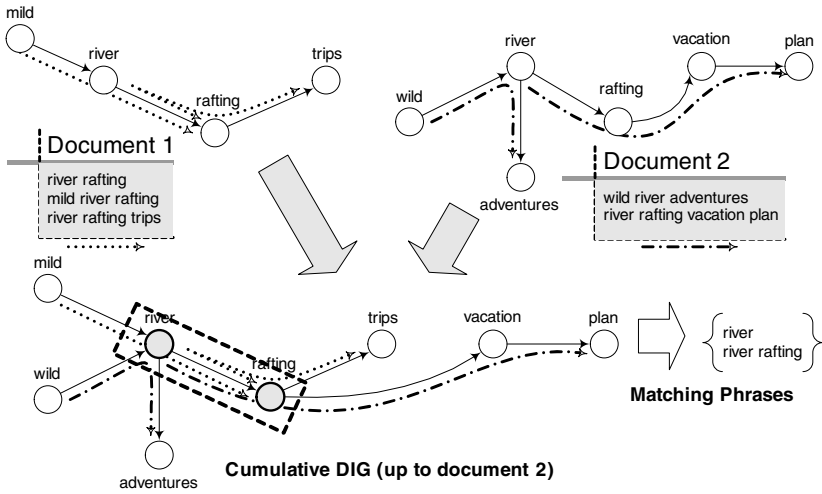


Fig. 2. Phrase Matching Using Document Index Graph

w : average **weight**; the average weight of the phrase over all documents. The weight of a phrase in a document is calculated using structural text cues. Examples: title phrases have maximum weight, section headings are weighted less, while body text is weighted lowest.

pf : average **phrase frequency**; the average number of times this phrase has appeared in one document, normalized by the length of the document in words.

$$pf = \arg \text{avg} \left[\frac{|\text{occurrences of } p|}{|\text{words in document}|} \right]$$

d : average phrase **depth**; the location of the first occurrence of the phrase in the document.

$$d = \arg \text{avg} \left[1 - \frac{|\text{words before first occurrence}|}{|\text{words in document}|} \right]$$

Those features will be used to rank the candidate phrases. In particular, we want phrases that appear in more documents (high df ; *i.e.* high *support*), have higher weights (high w), higher frequencies (high pf), and shallow depth (low d).

3.3 Phrase Ranking

Phrase features are used to calculate a *score* for each phrase. Phrases are then ranked by score, and a number of the top phrases are selected as the ones describing the topic of the cluster. There are two phrase scoring formulas used, as well as two methods of assigning the score to the candidate phrases, for a total of four variants of the CorePhrase algorithm.

First Scoring Formula. The score of each phrase p is:

$$\text{score}(p) = (w \cdot pf) \times -\log(1 - df) \quad (1)$$

The equation is derived from the $\text{tf} \times \text{idf}$ term weighting measure; however, we are rewarding phrases that appear in more documents (high df) rather than punishing those phrases. Notice also that the first scoring formula does not take the *depth* feature into account. We will refer to the variant of the algorithm that uses this formula as CorePhrase-1.

Second Scoring Formula. By examining the distribution of the values of each feature in a typical corpus (see Table 1 for details), it was found that the *weight* and *frequency* features usually have low values compared to the *depth* feature. To take this fact into account, it was necessary to “expand” the *weight* and *frequency* features by taking their square root, and to “compact” the *depth* by squaring it. This helps even out the feature distributions and prevents one feature from dominating the score equation. The formula is given in equation 2, and is used by the variant which we will refer to as CorePhrase-2.

$$\text{score}(p) = (\sqrt{w \cdot pf} \cdot d^2) \times -\log(1 - df) \quad (2)$$

Word weight-based score assignment. A *modified* score assignment scheme based on word weights is also used:

- First, assign *initial* scores to each phrase based on phrase scoring formulas given above.
- Construct a list of unique individual words out of the candidate phrases.
- For each word: add up all the scores of the phrases in which this word appeared to create a word weight.
- For each phrase: assign the *final* phrase score by adding the individual word weights of the constituent words and average them.

The variants of the algorithm that use this method will be referred to as CorePhrase-1M and -2M, based on the equation used to assign the phrase scores.

4 Experimental Results

4.1 Experimental Setup

Two data sets were used for evaluation, which are listed in Table 1¹. The first one is a collection of web documents representing six topics formed by submitting six different queries to the google search engine, and is used as a realistic evaluation in scenarios involving the online keyphrase extraction. The second data set is a collection of web documents from intranet web sites at the University of Waterloo, and serves in evaluating scenarios involving data-mining an intranet information system. Table 1 also lists the characteristics of each data set.

¹ The data sets are available for download at: <http://pami.uwaterloo.ca/~hammouda/webdata/>

Table 1. Data sets used for multi-document keyphrase extraction

class	# docs.	average words/doc.	candidate phrases	feature averages			
				<i>df</i>	<i>w</i>	<i>pf</i>	<i>d</i>
Data Set 1: Total 151 docs.							
canada transportation	22	893	24403	0.0934	0.2608	0.0007	0.4907
winter weather canada	23	636	5676	0.0938	0.2530	0.0019	0.5903
snowboarding skiing	24	482	990	0.0960	0.3067	0.0033	0.5215
river fishing	23	443	485	0.1129	0.2965	0.0042	0.5763
river rafting	29	278	599	0.0931	0.3680	0.0057	0.5612
black bear attacks	30	620	1590	0.0859	0.3593	0.0023	0.6024
Data Set 2: Total 161 docs.							
Co-operative Education	54	251	1379	0.0511	0.2672	0.0082	0.5693
Career Services	52	508	4245	0.0473	0.2565	0.0031	0.5000
Health Services	23	329	351	0.1454	0.2707	0.0057	0.5170
Campus Network	32	510	14200	0.0810	0.2569	0.0020	0.5198

A keyword-based extraction algorithm was used as a baseline for comparison. The algorithm extracts the centroid vector of a cluster represented as a set of keywords and selects the top frequent keywords in the cluster. This method is considered representative of most cluster labeling methods.

4.2 Evaluation Measures

Two extrinsic evaluation measures are used to assess the performance of CorePhrase. The first measure is called *overlap*, which measures the similarity between each extracted keyphrase to the predefined topic phrase of the cluster. The similarity is based on how many terms are shared between the two phrases. The overlap between an extracted keyphrase p_i and the topic phrase p_t is defined as:

$$\text{overlap}(p_i, p_t) = \frac{|p_i \cap p_t|}{|p_i \cup p_t|} \quad (3)$$

To evaluate the top k keyphrases as a set, we take the average overlap of the whole set. This measure is essentially telling us how well the top keyphrases, as a set, *fit* the reference topic.

The second evaluation measure is called *precision*², which gives an indication of how high the single keyphrase that best fits the topic is ranked. The best keyphrase is defined as the first keyphrase, in the top k , that has maximum overlap with the reference topic. Thus, the precision for the set of top k phrases (\mathbf{p}^k) with respect to the reference topic p_t is defined as:

$$\text{precision}(\mathbf{p}^k, p_t) = \text{overlap}(p_{\max}, p_t) \cdot \left[1 - \frac{\text{rank}(p_{\max}) - 1}{k} \right] \quad (4)$$

where $p_{\max} \in \mathbf{p}^k$ is the first phrase with maximum overlap in the top k phrases; and $\text{rank}(p_{\max})$ is its rank in the top k . In other words, precision tells us how *high* in the ranking the best phrase appears. The lower the best phrase comes in the ranking, the lower the precision.

² This is not the precision measure usually used in the information retrieval literature.

4.3 Discussion

Table 2 shows the results of keyphrase extraction by the CorePhrase algorithm variants for three of the classes (two classes from data set 1, and one class from data set 2)³. The phrases in the results are shown in stemmed form, with stop words removed.

A more concrete evaluation based on the quantitative measures, overlap and precision, is given in Table 3, which is illustrated also in Figure 3 (in the figure, only CorePhrase-2 and CorePhrase-2M are shown). For each of the four variants of the CorePhrase algorithm, in addition to the baseline keyword centroid algorithm, we report the overlap and precision. The average overlap is taken over the top 10 keyphrases/keywords of each cluster, with the maximum overlap value (best phrase) also shown.

Table 2. Keyphrase Extraction Results – Top 10 Keyphrases

CorePhrase-1	CorePhrase-2	CorePhrase-1M	CorePhrase-2M
canada transportation			
1 canada transport	canada transport	transport canada	canada transport
2 panel recommend	canada transport act	canada transport	transport canada
3 transport associ	transport act	road transport	transport act
4 transport associ canada	transport associ	transport issu	transport issu
5 associ canada	panel recommend	govern transport	recommend transport
6 canada transport act	unit state	surfak transport	transport polici canada transport
7 transport act	transport associ canada tac	public transport	canadian transport
8 road transport	associ canada tac	transport public	transport public
9 transport infrastructur	canada tac	transport infrastructur	public transport
10 transport associ canada tac	public privat sector	transport passeng	transport infrastructur
winter weather canada			
1 winter storm	sever weather	new hampshir new	environment assess environment
2 winter weather	winter weather	new jersey new	program legisl
3 environ canada	winter storm	new mexico new	program hunt
4 sever weather	weather warn	new hampshir new jersey new	fund program
5 weather warn	sever winter	new jersey new mexico new	environment link fund program
6 freez rain	sever weather warn	new hampshir new jersey new mexico	environment assess environment link fund
7 new brunswick	sever winter weather	new hampshir	environment link
8 heavi snowfal	new brunswick	hampshir new	environment assess environment link
9 winter weather warn	environ canada	carolina new hampshir new	assess environment
10 warn issu	cold winter	carolina new	environment assess
campus network			
1 campu network	campu network	network network	network network
2 uw campu network	uw campu network	network uw network	network level network
3 uw campu	uw campu	network level network	network uw network
4 roger watt	network connect	uw network	network subscrib network
5 roger watt ist	level network	network uw	level network level network
6 watt ist	high speed	network subscrib network	level network
7 ip address	uw resnet	network assign network	network level
8 ip network	connect uw	network uw campu network	network assign network
9 high speed	area campu network	network level	extern network level network level network
10 request registr	switch rout	level network level network	network level network rout

The keyphrases extracted by the variants of the CorePhrase⁴ algorithm were very close to the reference topic, which is a subjective verification of the algorithm correctness. We leave it to the reader to judge the quality of the keyphrases.

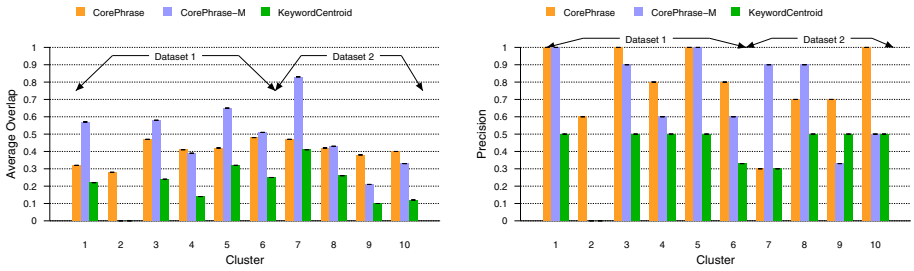
The first observation is that CorePhrase performs consistently better than the keyword centroid method. This is attributed to the keyphrases being in

³ Due to paper size limitation. A full list that can be obtained from: <http://pami.uwaterloo.ca/~hammouda/corephrase/results.html>

⁴ Throughout this discussion the name CorePhrase will refer to both CorePhrase-1 and CorePhrase-2, while CorePhrase-M will refer to both CorePhrase-1M and CorePhrase-2M; unless otherwise specified.

Table 3. Performance of the CorePhrase algorithm

class	CorePhrase-1		CorePhrase-2		CorePhrase-1M		CorePhrase-2M		Keyword Centroid	
	overlap (avg,max)	precision	overlap (avg,max)	precision	overlap (avg,max)	precision	overlap (avg,max)	precision	overlap (avg,max)	precision
Dataset 1										
canada transportation	(0.45,1.00)	1.00	(0.32,1.00)	1.00	(0.47,1.00)	1.00	(0.57,1.00)	1.00	(0.22,0.50)	0.5
winter weather canada	(0.22,0.67)	0.60	(0.28,0.67)	0.60	(0.00,0.00)	0.00	(0.00,0.00)	0.00	(0.00,0.00)	0.0
snowboarding skiing	(0.37,1.00)	1.00	(0.47,1.00)	1.00	(0.58,1.00)	0.90	(0.58,1.00)	0.90	(0.24,0.50)	0.5
river fishing	(0.41,1.00)	0.90	(0.41,1.00)	0.80	(0.43,1.00)	0.60	(0.39,1.00)	0.60	(0.14,0.50)	0.5
river rafting	(0.38,1.00)	1.00	(0.42,1.00)	1.00	(0.68,0.67)	0.90	(0.65,0.67)	1.00	(0.32,0.50)	0.5
black bear attacks	(0.45,1.00)	0.80	(0.48,1.00)	0.80	(0.47,1.00)	0.60	(0.51,1.00)	0.60	(0.25,0.33)	0.33
data set 1 average	(0.38,0.95)	0.88	(0.39,0.95)	0.87	(0.44,0.78)	0.67	(0.45,0.78)	0.68	(0.20,0.39)	0.39
Dataset 2										
co-operative education	(0.38,1.00)	0.20	(0.47,1.00)	0.30	(0.55,1.00)	0.80	(0.83,1.00)	0.90	(0.41,0.50)	0.3
career services	(0.37,1.00)	0.70	(0.42,1.00)	0.70	(0.58,1.00)	0.90	(0.43,1.00)	0.90	(0.26,0.50)	0.5
health services	(0.28,1.00)	0.70	(0.38,1.00)	0.70	(0.32,1.00)	0.50	(0.21,0.33)	0.33	(0.10,0.50)	0.5
campus network	(0.23,1.00)	1.00	(0.40,1.00)	1.00	(0.38,0.67)	0.20	(0.33,0.50)	0.50	(0.12,0.50)	0.5
data set 2 average	(0.31,1.00)	0.65	(0.42,1.00)	0.68	(0.46,0.92)	0.60	(0.45,0.71)	0.66	(0.22,0.46)	0.45
overall average	(0.35,0.97)	0.79	(0.40,0.97)	0.79	(0.45,0.83)	0.64	(0.45,0.75)	0.67	(0.21,0.42)	0.41



(a) Average Overlap

(b) Precision

Fig. 3. CorePhrase Accuracy Comparison

greater overlap with the reference topic than the naturally-shorter keywords. An interesting observation also is that CorePhrase-M, which is based on weighted words for phrase-scoring, and the keyword centroid follow the same trend. This is due to the link between the phrase scores and their constituent word scores.

Variants of the algorithm that use the depth feature are consistently better in terms of both overlap and precision. This is attributed to the fact that some common phrases usually appear at the end of each document (such as “last updated”, “copyright”, *etc.*). If depth information is ignored, these phrases make their way up the rank (*e.g.* the phrase “roger watt” in **campus network** cluster, which is the name of the network maintainer that appears at the end of each document.)

CorePhrase is somewhat better than its word-weighted counterpart (CorePhrase-M) in extracting the best phrase and ranking it among the top 10, where it achieves 97% overlap on average for the best phrase. The word-weighted variant achieves 83% maximum overlap on average for the best phrase.

When the top 10 keyphrases are considered as a *set*, the word-weighted variant achieves better average overlap performance in (45% for CorePhrase-M against 40% for CorePhrase). This is attributed to CorPhrase-M extracting heavily weighted words that often overlap with the topic, but not necessarily are the best descriptive phrases. (An example is the **winter weather canada** cluster.)

A final observation is that CorePhrase consistently achieves better precision than CorePhrase-M (79% for CorePhrase against 67% for CorePhrase-M.) This means that CorePhrase does not only find the best keyphrase, but ranks it higher than CorePhrase-M.

To summarize these findings: (a) CorePhrase is more accurate than keyword-based algorithms; (b) using phrase depth information achieves better performance; (c) using word-weights usually produces a better *set* of phrases; however, ignoring the word-weights usually produces better descriptive phrases; and (d) CorePhrase is able to identify the reference topic in the top few keyphrases.

5 Conclusions and Future Work

We presented the CorePhrase algorithm for accurately extracting descriptive keyphrases from text clusters. It is domain independent, and achieves high accuracy in identifying the topic of a document cluster compared with keyword-based cluster labeling algorithms.

The algorithm can be used to label clusters accurately, and to summarize clusters for other purposes such as calculating cluster-to-cluster or document-to-cluster similarity.

Future directions include using more features for the candidate phrases, based on heuristics of what constitutes a good phrase. Other ranking schemes are being investigated. Also the set of top keyphrases could be enhanced by removing spurious permutations and sub-phrases that appear in other phrases.

References

1. Turney, P.D.: Learning algorithms for keyphrase extraction. *Information Retrieval* **2** (2000) 303–336
2. Frank, E., Paynter, G., Witten, I., Gutwin, C., Nevill-Manning, C.: Domain-specific keyphrase extraction. In: 16th International Joint Conference on Artificial Intelligence (IJCAI-99), Stockholm, Sweden (1999) 668–673
3. Hammouda, K., Kamel, M.: Efficient phrase-based document indexing for web document clustering. *IEEE Transactions on Knowledge and Data Engineering* **16** (2004) 1279–1296
4. Neto, J., Santos, A., Kaestner, C., Freitas, A.: Document clustering and text summarization. In: Proc. 4th International Conference Practical Applications of Knowledge Discovery and Data Mining (PADD-2000), London, UK (2000) 41–55
5. Bakus, J., Kamel, M., Carey, T.: Extraction of text phrases using hierarchical grammar. In: *Advances in Artificial Intelligence: 15th Conference of the Canadian Society for Computational Studies of Intelligence (AI 2002)*, Calgary, Canada (2002) 319–324
6. Mani, I., Bloedorn, E.: Multi-document summarization by graph search and matching. In: *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-97)*, Providence, RI (1997) 622–628
7. Clifton, C., Cooley, R., Rennie, J.: TopCat: data mining for topic identification in a text corpus. *IEEE Transactions on Knowledge and Data Engineering* **16** (2004) 949–964

A New Multidimensional Feature Transformation for Linear Classifiers and Its Applications

EunSang Bak

Electrical and Computer Engineering Department,
University of North Carolina,
9201 University City Blvd, Charlotte, NC 28223, U.S.A
bakeunsang@yahoo.com

Abstract. In this paper, a new feature transformation method is introduced to decrease misclassification rate. Linear classifiers in general are not able to classify feature vectors which lie in a high dimensional feature space. When the feature vectors from difference classes have underlying distributions which are severely overlapped, it is even more difficult to classify those feature vectors with desirable performance. In this case, data reduction or feature transformation typically finds a feature subspace in which feature vectors can be well separated. However, it is still not possible to overcome misclassifications which results from the overlapping area. The proposed feature transformation increases the dimension of a feature vector by combining other feature vectors in the same class and then follows typical data reduction process. Significantly improved separability in terms of linear classifiers is achieved through such a sequential process and is identified in the experimental results.

1 Introduction

The purpose of pattern classification is to decide the class of the data which is assumed to consist of (C, \mathbf{x}) pairs where C is the class to which \mathbf{x} , which is an r -dimensional feature vector, belongs. There are many traditional and modern approaches to estimate the underlying distributions. Some of them [1], [8] attempt to estimate the class conditional probability distribution of a feature vector \mathbf{x} by assuming specific distributional form of underlying distributions. On the contrary, other approaches [2], [4], [5], which are typically called nonparametric methods, try to estimate the probability distributions without assuming any distributional form.

Once the distributions are estimated in either method, feature space is separated by a selected classifier. In fact, the selection of classifier is strongly related with the estimation method since the classifier is built upon the estimated underlying distributions.

In a high dimensional feature space, classification process may suffer from so called curse of dimensionality. Most of the approaches for feature classification involve a data reduction or feature transformation step. This step basically reduces the dimension of feature space so that feature vectors can be well separated in the new lower dimensional feature space.

Various methods have been proposed in literature for feature transformation/data reduction; Fisher's approach [3], [6], [12], [13], removal classification structures [15], adaptive linear dimensionality reduction [16], linear constrained distance-based classifier analysis [17] and others [7], [8], [9], [10], [11]. These approaches consistently try to transform feature vectors into a feature space of lower dimension.

In this paper, a new feature transformation is proposed and its effect is investigated in terms of linear classifiers. The paper is organized as follows: In Section 2, basic idea of the proposed feature transformation is introduced. In Section 3, a new feature transformation is explained in detail with mathematical derivation and its effect on the underlying distributions is analyzed. Unsupervised image classification process incorporating a new feature transformation is described in Section 4 as an application. The experimental results are presented in Section 5, which is followed by the conclusions in Section 6.

2 Basic Idea

For solving multidimensional classification problems, Fisher [3] suggested a method, which projected the data from d dimensions onto a line with specific direction for which the projected data were well separated. The projected value was obtained by linear combination of the components of \mathbf{x} , thus every multidimensional data was converted to a scalar by this linear combination.

Such a linear combination turned out to be the product of the difference of the means of classes and the common inverse covariance matrix. This process is equivalent to maximizing the ratio of between-class variation to within-class variation.

Fisher's main idea would rather be interpreted as how to linearly reduce the dimension of the feature space so that linear classifiers can give the best classification result. In this paper, the proposed method takes an opposite direction. Instead of reducing the dimension of the data, we first increase the dimension of the data by combining several feature vectors in the same class and make a new feature space and then reduce the dimension of the new feature space. By inserting the process of increasing data dimension, a very interesting fact is found with respect to feature classification.

Comparing the distributions between the original feature space and a new feature space, a new feature space gives much better separability for classification. That is, the distance between the means of existing probability distributions in the feature space gets longer compared to the change of their standard deviations. This in turn reduces the overlapping area between the probability distributions. Such an augmentation of the dimension of the feature space will be called a feature transformation hereafter.

Feature transformation requires a couple of assumptions. One of them is that the probability distributions of classes in the feature space should be normally distributed and the other is that those distributions have a common covariance.

In the following section, the proposed feature transformation will be derived mathematically and shows how the feature transformation changes the class conditional probability distributions so as to be suitable for linear classifiers.

3 Multivariate Feature Transformation

Suppose there are two classes and the i^{th} class is represented as $\pi_i \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma})$, $i=1,2$. It is assumed that all the classes have a common covariance matrix $\boldsymbol{\Sigma}$. We would like to assign a sample data \mathbf{x} , which is an r -dimensional random vector, to the most probable class among the two classes.

Let us define a new random vector \mathbf{x}^* which consists of a certain number of random vectors in the same class. In other words, a new random vector is a composite random vector whose elements are random vectors from the same class. For example, take $(s + 1)$ random vectors around a neighborhood of the centered random vector \mathbf{x}_0 , $\{\mathbf{x}_j^{(i)} : j=0, \dots, s\}$ in the i^{th} class and make a new random vector $\mathbf{x}_0^{(i)*}$ which has $r(s + 1)$ dimension in proportion to the number of component vectors. The random vector $\mathbf{x}_0^{(i)*}$ is regarded as an extended random vector of $\mathbf{x}_0^{(i)}$ and is represented as follows:

$$\mathbf{x}^{(i)*} = \left[\mathbf{x}_0^T \ \mathbf{x}_1^T \ \dots \ \mathbf{x}_s^T \right]^T \tag{1}$$

Note that the subscript of $\mathbf{x}^{(i)*}$ is omitted for brevity. Unless otherwise specified, the subscript of the extended vector is the same as that of the first component vector. The mean vector of $\mathbf{x}^{(i)*}$ whose component random vectors are from the i^{th} class becomes

$$\boldsymbol{\mu}_i^* = \mathbf{1} \otimes \boldsymbol{\mu}_i \tag{2}$$

where the operator \otimes denotes a direct product in matrix operation. In addition, a covariance between two random vectors ($\mathbf{x}_j^{(i)}$, $\mathbf{x}_k^{(i)}$) can be defined as

$$\text{Cov}(\mathbf{x}_j^{(i)}, \mathbf{x}_k^{(i)}) = \rho_{jk} \boldsymbol{\Sigma} \tag{3}$$

which generate the covariance matrix of $\mathbf{x}^{(i)*}$ in (4). For simplicity, the superscript for labeling the class from which the random vectors come is omitted, therefore, \mathbf{x}_j is substituted for $\mathbf{x}_j^{(i)}$ and \mathbf{x}^* is substituted for $\mathbf{x}^{(i)*}$ as needed.

$$\text{Cov}(\mathbf{x}^*) = \begin{bmatrix} \text{Cov}(\mathbf{x}_0, \mathbf{x}_0) & \dots & \text{Cov}(\mathbf{x}_0, \mathbf{x}_s) \\ \vdots & & \vdots \\ \text{Cov}(\mathbf{x}_s, \mathbf{x}_0) & \dots & \text{Cov}(\mathbf{x}_s, \mathbf{x}_s) \end{bmatrix} = \mathbf{C}^* \otimes \boldsymbol{\Sigma} = \boldsymbol{\Sigma}^* \tag{4}$$

As a result, a new multivariate normal random vector \mathbf{x}^* whose mean vector is $\boldsymbol{\mu}^*$ and covariance matrix $\boldsymbol{\Sigma}^*$ is obtained.

$$\mathbf{x}^* \sim N(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) \tag{5}$$

Now, discriminant function D_i^* [18] for the random vector $\mathbf{x}^{(i)*}$ is written compared to D_i for $\mathbf{x}^{(i)}$.

$$\begin{aligned} D_i &= \mathbf{L}_i^T \mathbf{x}^{(i)} - \frac{1}{2} \mathbf{L}_i^T \boldsymbol{\mu}_i, \quad \mathbf{L}_i^T = (\boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_i)^T \\ D_i^* &= \mathbf{L}_i^{*T} \mathbf{x}^{(i)*} - \frac{1}{2} \mathbf{L}_i^{*T} \boldsymbol{\mu}_i^*, \quad \mathbf{L}_i^{*T} = (\boldsymbol{\Sigma}^{*-1} \boldsymbol{\mu}_i^*)^T \end{aligned} \tag{6}$$

After some mathematical expansions, \mathbf{L}_i^* can be described in terms of \mathbf{L}_i in (7). The δ in (7) is a vector whose elements are the sum of elements of each row vector of the inverse of correlation coefficient matrix \mathbf{C}^* in (4).

$$\begin{aligned} \mathbf{L}_i^{*T} &= \delta^T \otimes \mathbf{L}_i^T \\ \delta &= (\mathbf{1}^T \mathbf{C}^{*-1})^T = [\delta_0, \delta_1, \dots, \delta_s]^T \end{aligned} \tag{7}$$

The discriminant function D_i^* is finally represented in (8) and is described in terms of the random vector \mathbf{x} . In other words, the discriminant function of composite random vector \mathbf{x}^* is characterized by the terms in the discriminant function of the random vector \mathbf{x} .

$$D_i^* = \mathbf{L}_i^T \left(\sum_{n=0}^s \delta_n \mathbf{x}_n^{(i)} - \frac{1}{2} \boldsymbol{\mu}_i \sum_{n=0}^s \delta_n \right) \tag{8}$$

Now, we define a new random vector \mathbf{G} in (9) as a linear combination of $(s + 1)$ number of random vector \mathbf{x} 's.

$$\mathbf{G} = \sum_{n=0}^s \delta_n \mathbf{x}_n \tag{9}$$

Eq. (9) corresponds to the proposed feature transformation. The coefficients of the linear combination are derived from the correlation coefficient matrix of random vector \mathbf{x} . Once the expectation $\boldsymbol{\mu}_G$ and the covariance matrix $\boldsymbol{\Sigma}_G$ are obtained, the discriminant function of \mathbf{G} is described in (10). Comparing (10) with (8), it is recognized that the results are equivalent, which means that the discriminant function D_i^* can be considered as the discriminant function $D_i(\mathbf{G})$.

$$\begin{aligned} D_i(\mathbf{G}) &= (\boldsymbol{\Sigma}_G^{-1} \boldsymbol{\mu}_G)^T \mathbf{G} - \frac{1}{2} (\boldsymbol{\Sigma}_G^{-1} \boldsymbol{\mu}_G)^T \boldsymbol{\mu}_G \\ &= \mathbf{L}_i^T \left[\sum_{n=0}^s \delta_n \mathbf{x}_n - \frac{1}{2} \boldsymbol{\mu}_i \sum_{n=0}^s \delta_n \right] \end{aligned} \tag{10}$$

Since a linear combination of the normal random vectors also follows a normal distribution, the distribution of the random vector \mathbf{G} will be

$$\begin{aligned} \mathbf{G} &\sim N(\boldsymbol{\mu}_G, \boldsymbol{\Sigma}_G) \\ \boldsymbol{\mu}_G &= k \cdot \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_G = k \cdot \boldsymbol{\Sigma}, \quad k = \sum_n \delta_n \end{aligned} \tag{11}$$

In summary, a multivariate random vector \mathbf{x} is transformed into a multivariate random vector \mathbf{G} by means of creating a composite random vector \mathbf{x}^* . The mean vector of \mathbf{G} simply becomes k times larger than the mean vector of \mathbf{x} and the covariance matrix becomes also k times larger than that of \mathbf{x} . This is a very important observation to indicate how feature transformation changes the class conditional probability distributions and achieves better separability. Since the distance between the means

becomes k times larger while the spread of the distributions becomes \sqrt{k} times larger, the distributions get farther away after feature transformation and it gives better separability.

4 Unsupervised Image Classification

In the previous section, we have seen that the linear combination of an extended random vector whose component random vectors came from the same class produced a new random vector and the separability of new random vectors between different classes was significantly improved in terms of linear classifiers.

One of the applications for the proposed method would be image classification. In general, objective of image classification is the separation of the regions or objects in the image which have different characteristics. Due to the characteristics of image, most of the homogeneous objects or regions occupy a certain area in an image and feature vectors from the same object are located in the neighborhood. Thus, determination of classification of a feature vector can be made associated with the determinations of the neighboring feature vectors.

In our experiments, an iterative unsupervised classification is chosen. This classification process does not need a training process. Given that the number of regions (or classes) into which an image is supposed to be separated, a simple clustering algorithm is applied to classify the feature vectors and thus a provisional result of classification is obtained.

The purpose of the provisional result of classification by given clustering algorithm is to extract intermediate information about each class and to calculate the coefficient vectors δ for each class for feature transformation, which compose an initial iteration. From the second iteration a selected linear classifier classifies the feature space resulted from the previous feature transformation.

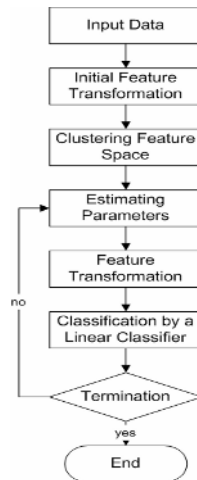


Fig. 1. Unsupervised classification procedure

As the proposed feature transformation has been mathematically proved in Section 3, the distributions of feature vectors of the classes becomes farther away each other, which results in a smaller misclassification rate from the smaller overlapping area between two probability distributions.

Until the conditions for terminating process are satisfied, the iterative procedure is continued. One of the conditions for termination is the size of the value k in (13). If k is not larger than one, such a transformation does not give a better separability in the feature space.

Fig. 1 shows the iterative procedure for image classification explained in the above. Note that since the main contribution of this paper is the method of feature transformation, any other clustering method can be used depending on particular need, although, in our experiments, K-means clustering method is used.

5 Experimental Results

We simulate the proposed method with a synthetic data set. The data set is generated from multivariate number generator with a mean vector and a covariance matrix. Each feature vector is located by given coordinate in an image as Fig. 2(e) according to its class. The size of the image is 26x26 so that the total number of feature vectors from two classes is 676. Synthetic feature vectors are generated with parameter sets whose mean vectors are [-1.0 1.0] and [1.0 -1.0] each, and the common covariance matrix whose diagonal elements are 10.0.

Fig. 2(a) shows the initial distributions of feature vectors from two classes. Since the distance from two mean vectors is much smaller than the size of variances in the covariance matrix, the feature vectors are heavily overlapped between classes. None of the linear classifiers seem to be adequate to classify feature vectors with desirable performance. Assuming that feature vectors are independently extracted, the first feature transformation is executed with the δ vector having all ones. After first transformation, K-means clustering method is used to make a temporary classification map. Fig. 2(b) shows this classification map and the two classes in Fig. 2(b) look more separated than in Fig. 2(a). Fig. 2(b) simply represent temporary determined classes so that each class may contain feature vectors that are actually misclassified.

After four iterations, the distributions of feature vectors become more suitable for linear classification as in Fig. 2(c). Now, any linear classifiers can be selected to separate the transformed feature vectors. Fig. 2(d) shows the error rate at each iteration by linear discriminant function and the locations of misclassified feature vectors are illustrated in Fig. 2(f) compared to the true classification map in Fig. 2(e). In the light of the above simulations, the proposed method indicates a new possibility for image classification.

Now, the proposed method is applied to a practical classification data. Original data set is from the UCI Machine Learning Repository [14]. Feature vectors in the original data set are extracted from an image which contains seven different regions, which are grass, path, cement, sky, brickface, foliage and window. A feature vector from each region is characterized by 19 feature components. In our experiments, for

the sake of visualization, two regions (cement and foliage) are selected and two most significant feature components (intensity-mean and value-mean) are chosen in the experiments.

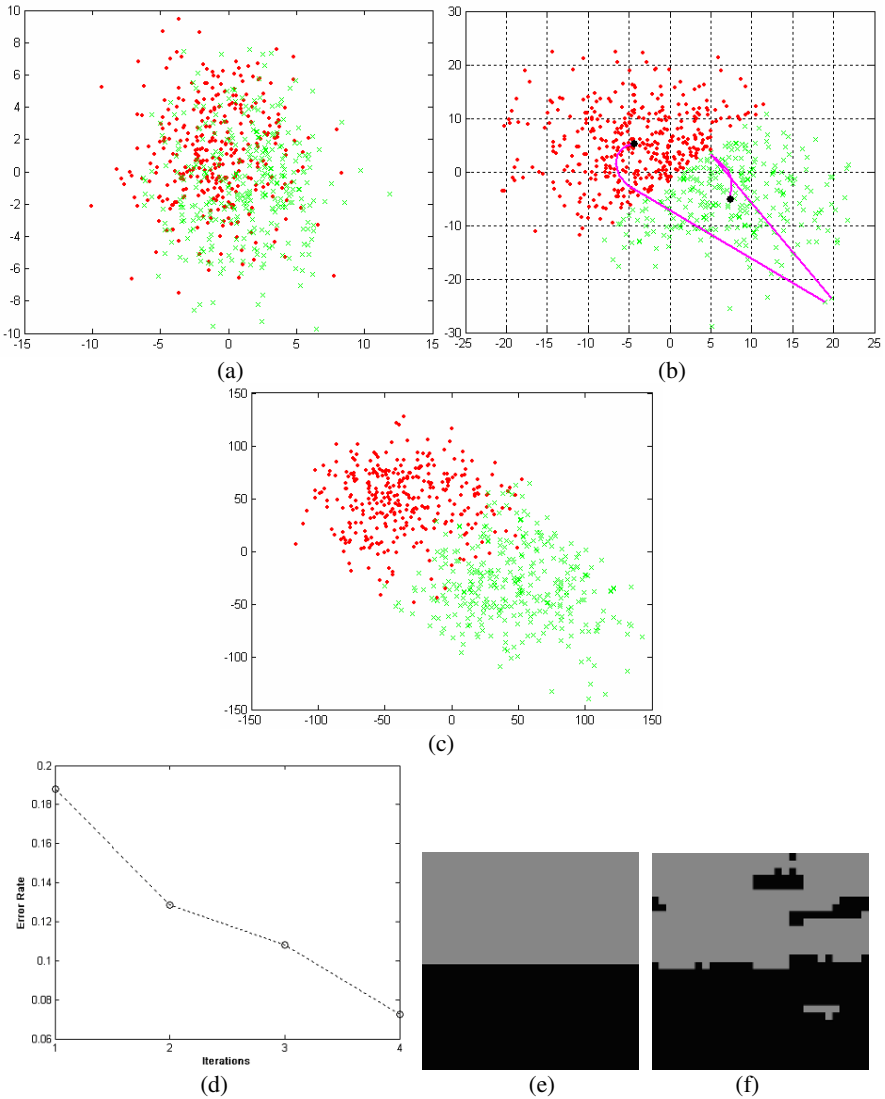


Fig. 2. Result from simulation. (a) Distributions of feature vectors of two classes. (b) Temporary classification by K-means algorithm. (c) Distributions of feature vectors of two classes after the last feature transformation. (d) Error rates on every iteration. (e) True classification map. (f) Classification map from the proposed method

The data seems to be relatively linearly separable. However, as can be in Fig. 3(a) and Fig. 3(b), data distributions are not completely separable so that it would not be possible to separate the data without misclassification by any linear classifiers.

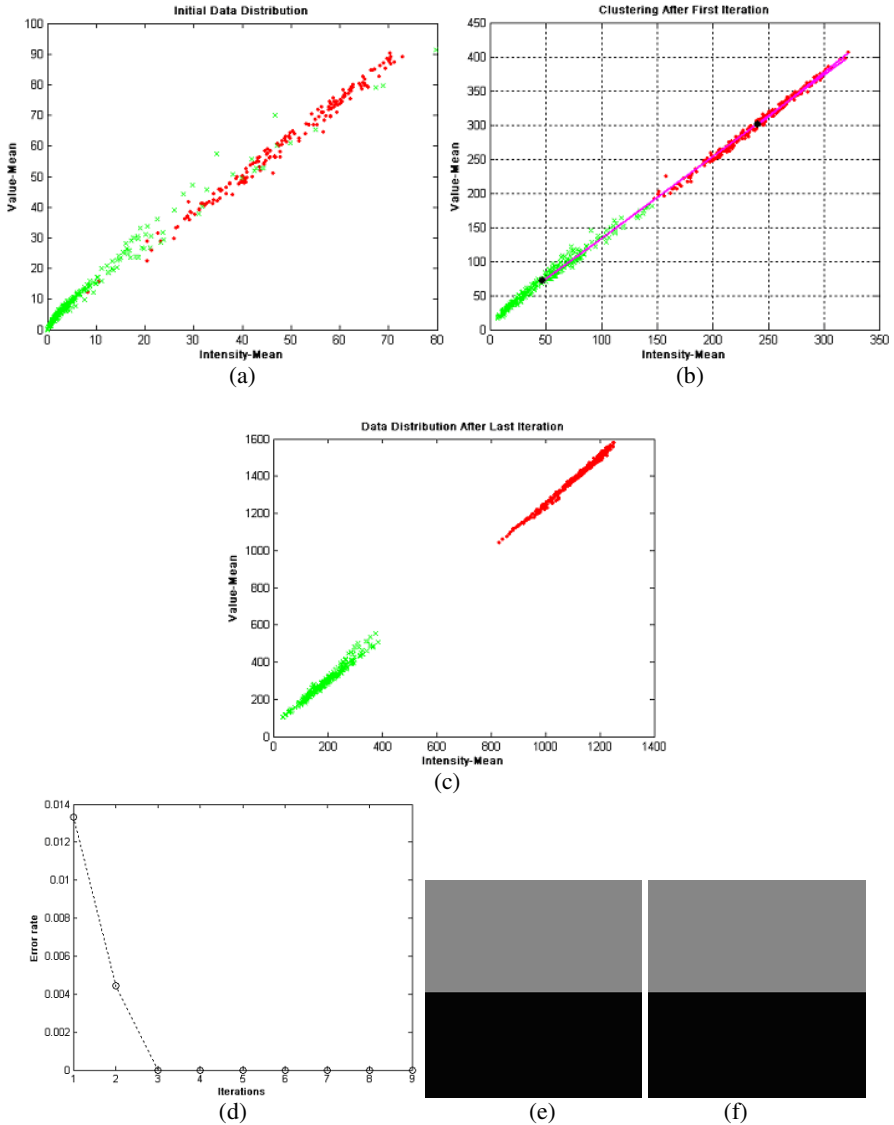


Fig. 3. Results from a real data set. (a) Distributions of feature vectors of two classes. (b) Temporary classification by K-means algorithm. (c) Distributions of feature vectors of two classes after the last feature transformation. (d) Error rates on every iteration. (e) True classification map. (f) Classification map from the proposed method

Surprisingly, Fig. 3(c)-(f) show the results of classification using feature transformation. Taking into account the features in the neighborhood, the proposed method changes the data distributions as much as it can be linearly separated. Fig. 3(c) shows the last data distributions on which linear discriminant function is to be applied. Feature vectors are separated without misclassifications and are illustrated in Fig. 3(f). As a result, two regions (cement-foliage) which have different natural characteristics are completely separated without misclassification through the proposed method.

6 Conclusions

A new feature transformation method is introduced. It increases the dimension of a feature vector by combining other feature vectors in the same class and then follows a typical data reduction process. The proposed method eventually gives significantly improved separability in feature space in terms of linear classifiers and the promising experimental results are presented.

References

1. W. Chou, "Discriminant-function-based minimum recognition error rate pattern-recognition approach to speech recognition," *Proc. IEEE*, vol. 88, no. 8, pp. 1201-1223, Aug. 2000.
2. T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 6, pp. 607-616, June 1996.
3. R.A. Fisher, "The statistical utilization of multiple measurements," *Annals of Eugenics*, vol. 8, pp. 376-386, 1938.
4. L.J. Buturovic, "Towards Bayes-optimal linear dimension reduction," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, pp. 420-424, 1994.
5. T. Hastie and R. Tibshirani, "Discriminant analysis by Gaussian mixtures," *J. Royal Statistical Soc., B*, vol. 58, pp. 155-176, 1996.
6. C.R. Rao, "The utilization of multiple measurements in problems of biological classification," *J. Royal Statistical Soc., B*, vol. 10, pp. 159-203, 1948.
7. M. Hubert and K.V. Driessen, "Fast and robust discriminant analysis," *Computational Statistics & Data Analysis*, vol. 45, Issue 2, pp. 301-320 March 2004.
8. W.L. Poston and D.J. Marchette, "Recursive dimensionality reduction using Fisher's linear discriminant," *Pattern Recognition*, vol. 31, no. 7, pp. 881-888, 1998.
9. M. Loog, R.P.W. Duin and R. Haeb-Umbach, "Multiclass linear dimension reduction by weighted pairwise Fisher criteria," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 7, pp. 762-766, 2001.
10. L. Rueda and B.J. Oommen, "On optimal pairwise linear classifiers for normal distributions: The two-dimensional case," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 274-280, 2002.
11. H. Brunzell and J. Eriksson, "Feature reduction for classification of multidimensional data," *Pattern Recognition*, vol. 33, pp. 1741-1748, 2000.
12. Duda, R.O., P.E Hart, and D.G.. Stork, *pattern Classification*, 2ed. John Wiley & Sons, New York, Jan. 2000.
13. A.K. Jain, R.P.W. Duin and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 4-37, 2000.

14. UCI Repository of Machine Learning Databases, www.ics.uci.edu/mllearn/mlrepository.html, 2004
15. M. Aladjem, "Linear discriminant analysis for two classes via removal of classification structure," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 187-192, 1997
16. R. Lotlikar and R. Kothari, "Adaptive linear dimensionality reduction for classification," *Pattern Recognition*, vol. 33, Issue 2, pp. 177-350 2000
17. Q. Du and C.-I. Chang, "A linear constrained distance-based discriminant analysis for hyperspectral image classification," *Pattern Recognition*, vol. 34, Issue 2, pp. 361-373, Feb. 2001
18. A.M. Kshirsagar, *Multivariate Analysis*, M. Dekker, New York, 1972.

Comparison of FLDA, MLP and SVM in Diagnosis of Lung Nodule

Aristófanés Corrêa Silva¹, Anselmo Cardoso de Paiva²,
and Alexandre Cesar Muniz de Oliveira²

¹ Federal University of Maranhão - UFMA, Department of Electrical Engineering,
Av. dos Portugueses, SN, Campus do Bacanga, Bacanga,
65085-580, São Luís, MA, Brazil
ari@dee.ufma.br

² Federal University of Maranhão - UFMA, Department of Computer Science,
Av. dos Portugueses, SN, Campus do Bacanga, Bacanga,
65085-580, São Luís, MA, Brazil
{paiva, acmo}@deinf.ufma.br

Abstract. The purpose of the present work is to compare three classifiers: Fisher's Linear Discriminant Analysis, Multilayer Perceptron and Support Vector Machine to diagnosis of lung nodule. These algorithms are tested on a database with 36 nodules, being 29 benigns and 7 malignants. Results show that the three algorithms had similar performance on this particular task.

Keywords: Fisher's Linear Discriminant Analysis, Multilayer Perceptron, Support Vector Machine, Diagnosis of Lung Nodule.

1 Introduction

Lung cancer is known as one of the cancers with shortest survival after diagnosis [1]. Therefore, the sooner it is detected the larger the patient's chance of cure. On the other hand, the more information physicians have available, the more precise the diagnosis will be.

Solitary lung nodules are an approximately round lesion less than 3 cm in diameter and completely surrounded by pulmonary parenchyma. Larger lesions should be referred to as pulmonary masses and should be managed as likely malignant. In this situation, prompt diagnosis and staging are necessary to choose the proper treatment [1].

On the other hand, manage a pulmonary nodule is always a challenge, because in spite of benign possibility becoming more important as the nodule's dimension decreases, malignity has always to be excluded. If malignity has more than 5% of chance to be present a biopsy method must be indicated and for more than 60% the patient is sent directly to resection. Less than 5% allows a close following to prove stability. A recent problem, which has been becoming more frequent nowadays, is that Computerized Tomography (CT) is finding

nodules not visible in a conventional chest X-ray in high risk groups (ie. heavy smokers) and frequently their little dimensions make difficult or impossible a biopsy procedure. On the other side, systematic resection would increase unnecessary surgery at unacceptable levels. In this circumstance, a quantitative image method of volumetric determination are becoming recognized as an important parameter to establish following criteria.

Macroscopically, lung nodules have a very variable tissue's structure. There can be nodules with tissue alterations almost imperceptible to the human eye and others presenting very noticeable alterations. The diagnosis gold standard is the histological examination, but image methods and in special Computed X-ray Tomography can aid diagnostic process in analyzing nodule's attributes like shape, presence and pattern of calcifications, walls of cavitations, aerobronchogram and, more recently, mean attenuation coefficient before and after intravenous contrast standardized injection.

However, besides numerous reports of qualitative morphologic CT data in medical literature, there are relatively few reports of quantitative CT data and it seems that, in general, they are underutilized. We hypothesized that quantitative CT data derived from geometric and texture parameters may contribute to differential diagnosis between benign and malignant solitary pulmonary nodule, even without contrast utilization. The top row in Figure 1 shows the texture for two benign (a and b) and two malignant (c and d) nodules. The bottom row in Figure 1 shows the shape for two benign (a and b) and two malignant (c and d).

The purpose of the present work is to compare three classifiers: Fisher's Linear Discriminant Analysis, Multilayer Perceptron and Support Vector Machine to diagnosis of lung nodule. Features extracted of nodules are based on CT images and analysis is supplied regarding the 3D geometry of the nodule. The validation of the classifiers is done by means of leave-one-out technique. The analysis and evaluation of tests was done using the area under the ROC (Receiver Operation Characteristic) [2] curve.









	Benign		Malignant	
Label	(a)	(b)	(c)	(d)
1 Slice				
2 3D Reconstruction				

Fig. 1. Examples of benign lung nodules and malignant lung nodules

2 Methods

2.1 Image Acquisition

The images were acquired with a Helical GE Pro Speed tomography under the following conditions: tube voltage 120 kVp, tube current 100 mA, image size 512×512 pixels, voxel size $0.67 \times 0.67 \times 1.0$ mm. The images were quantized in 12 bits and stored in the DICOM format [3].

2.2 3D Extraction and Reconstruction of Lung Nodules

In most cases, lung nodules are easy to be visually detected by physicians, since their shape and location are different from other lung structures. However, the nodule's voxel density is similar to that of other structures, such as blood vessels, which makes automatic computer detection difficult. This happens especially when a nodule is adjacent to the pleura. For these reasons, we have used the 3D region-growing algorithm with voxel aggregation [4] to make the nodule detection, which provides physician greater interactivity and control over the segmentation and determination of required parameters (thresholds, initial and final slice, and seed).

Two other resources provide greater control in the segmentation procedure: the barrier and the eraser. The barrier is a cylinder placed around the nodule by the user with the purpose of restricting the region of interest and stopping the segmentation by voxel aggregation from invading other lung structures. The eraser is a resource of the system that allows physicians to erase undesired structures, either before or after segmentation, in order to avoid and correct segmentation errors [5]. The bottom row in Figure 1 shows the 3D reconstruction of the nodules in the top row and exemplifies the nodule segmentation.

2.3 Lung Nodule Features

Skeletonization is a convenient tool to obtain a simplified representation of shapes that preserves most topological information [6]. A skeleton captures the local symmetry axes and is therefore centered in the object. In image analysis, features extracted from skeletons are commonly used in pattern-recognition algorithms [7]. Skeletons contain information about shape features which are very important in this work context.

We have used Zhou and Toga's algorithm [8] in the skeletonization process. They have proposed a voxel-coding approach to efficiently skeletonize volumetric objects. Each object point has two codes. One is the Boundary Seeded code (BS), which coincides with the traditional distance transform to indicate the minimum distance to the object's boundary. The second code is the so-called Single Seeded code (SS), which indicates the distance to a specific reference point. SS code is used to extract the shortest path between a point in the object and the reference point. These paths are represented by sequential sets of voxels that will compose the initial skeleton. The key idea of voxel coding is to use the SS codes to generate a connected raw skeleton and the BS codes to assure the centeredness of the final skeleton.

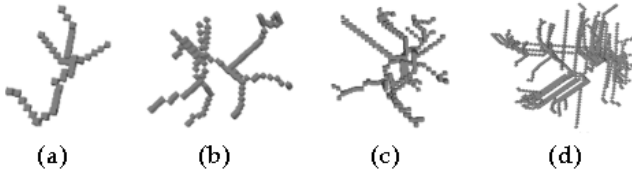


Fig. 2. Application of the skeleton algorithm based on the nodules in Figure 1(a), (b), (c) and (d)

Figure 2 shows the application the skeleton algorithm based on the nodules in Figure 1(a), (b), (c) and (d), respectively. It is easy observe that malignant nodules have more segments than benign nodules.

We have extracted ten measures based on skeletons to analyze lung nodules, six of them have been used to describe the nodule’s geometry. They are :

- a) Number of Segments (NS)
- b) Number of Branches (NB)
- c) Fraction of Volume (FV): FV is defined by

$$FV = \frac{v}{V}$$

where v is the skeleton volume and V is the lung nodule’s volume.

- d) Length of Segments (LS):Defined by

$$LS = \frac{L}{\sqrt[3]{V}}$$

where L is the length of all segments and V is the lung nodule’s volume.

- e) Volume of the Skeleton Convex Hull (VCH)
- f) Rate between the number of segments and the volume of the convex hull (NSVCH) [7]

Trying to charcaterize the nodule based on the skeleton texture we compute the density histogram of the N larger skeleton segments, where N is the smaller number of segments in the nodule’s database. From this histogram e compute:

- g) Variation Coefficient (VC): The VC is a measure of relative dispersion and is given by

$$VC = \frac{\sigma}{\mu}$$

where σ is the standard deviation and μ is the mean.

- h) Histogram Moments (variance (M_2), skewness (M_3), kurtosis (M_4)) defined as:

$$M_n = \frac{\sum (x_i - \mu)^n f_i}{N} \tag{1}$$

where $n = 2, 3, 4$, μ is the mean, N denotes the number of voxels in the segment, and f_i is the histogram.

More detailed information on moment theory can be found in [9].

2.4 Classification Algorithms

A wide variety of approaches has been taken towards the classification task. Three main historical strands of research can be identified [10]: statistical, neural network and machine learning. This section give an overview of Fisher’s Linear Discriminant Analysis, Multilayer Perceptron and Support Vector Machine based on paradigms cited above.

Fisher’s Linear Discriminant Analysis - FLDA: Linear discrimination, as the name suggests, looks for linear combinations of the input variables that can provide an adequate separation for the given classes. Rather than look for a particular parametric form of distribution, LDA uses an empirical approach to define linear decision planes in the attribute space i.e. it models a surface. The discriminant functions used by LDA are built up as a linear combination of the variables that seek to somehow maximize the differences between the classes [11]:

$$y = \beta_1x_1 + \beta_2x_2 + \dots + \beta_nx_n = \beta' x \tag{2}$$

The problem then reduces to find a suitable vector β . There are several popular variations of this idea, one of the most successful being the Fisher Linear Discriminant Rule. Fisher’s Rule is considered a “sensible” classification, in the sense that it is intuitively appealing. It makes use of the fact that distributions that have a greater variance between their classes than within each class should be easier to separate. Therefore, it searches for a linear function in the attribute space that maximizes the ratio of the between-group sum-of-squares (B) to the within-group sum-of-squares (W). This can be achieved by maximizing the ratio

$$\frac{\beta' B \beta}{\beta' W \beta} \tag{3}$$

and it turns out that the vector that maximizes this ratio, β , is the eigenvector corresponding to the largest eigenvalue of $W^{-1}B$ i.e. the linear discriminant function y is equivalent to the first canonical variate. Hence the discriminant rule can be written as:

$$x \in i \text{ if } |\beta^T x - \beta^T u_i| < |\beta^T x - \beta^T u_j|, \text{ for all } j \neq i \tag{4}$$

where $W = \sum n_i S_i$ and $B = \sum n_i (x_i - x)(x_i - x)'$, and n_i is class i sample size, S_i is class i covariance matrix, x_i is the class i mean sample value and x is the population mean.

Stepwise discriminant analysis [11] was used to select the best variables to differentiate between groups. These measures were used in the FLDA, MLP and SVM classifiers.

Multilayer Perceptron: The Multilayer Perceptron - MLP, a feed-forward back-propagation network, is the most popular neural network technique in pattern recognition [12], [13]. Briefly, MLPs are supervised learning classifiers that consist of an input layer, an output layer, and one or more hidden layers that

extract useful information during learning and assign modifiable weighting coefficients to components of the input layers. In the first (forward) pass, weights are assigned to the input units and to the nodes in the hidden layers and between the nodes in the hidden layer and the output, determine the output. The output is compared with the target output. An error signal is back propagated and the connection weights are adjusted correspondingly. During training, MLPs construct a multidimensional space, defined by the activation of the hidden nodes, so that the two classes (benign and malignant nodules) are as separable as possible. The separating surface adapts to the data.

Support Vector Machine: The Support Vector Machine (SVM) introduced by V. Vapnik in 1995 is a method to estimate the function classifying the data into two classes [14], [15]. The basic idea of SVM is to construct a hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized. The SVM term comes from the fact that the points in the training set which are closest to the decision surface are called support vectors. SVM achieves this by the structural risk minimization principle that is based on the fact that the error rate of a learning machine on the test data is bounded by the sum of the training-error rate and a term that depends on the Vapnik-Chervonenkis (VC) dimension.

The process starts with a training set of points $x_i \in \mathfrak{R}^n, i = 1, 2, \dots, l$ where each point x_i belongs to one of two classes identified by the label $y_i \in \{-1, 1\}$. The goal of maximum margin classification is to separate the two classes by a hyperplane such that the distance to the support vectors is maximized. The construction can be thought as follow: each point x in the input space is mapped to a point $z = \Phi(x)$ of a higher dimensional space, called the feature space, where the data are linearly separated by a hyperplane. The nature of data determines how the method proceeds. There are data that are linearly separable, nonlinearly separable and with impossible separation. This last case be still tracted by the SVM. The key property in this construction is that we can write our decision function using a kernel function $K(x, y)$ which is given by the function $\Phi(x)$ that maps the input space into the feature space. Such decision surface has the equation:

$$f(x) = \sum_{i=1}^l \alpha_i y_i K(x, x_i) + b \quad (5)$$

where $K(x, x_i) = \Phi(x) \cdot \Phi(x_i)$, and the coefficients α_i and the b are the solutions of a convex quadratic programming problem [14], namely

$$\begin{aligned} \min_{w, b, \xi} \quad & \frac{1}{2} w^T \cdot w + C \sum_{i=1}^l \xi_i \\ \text{subject to} \quad & y_i [w^T \cdot \phi(x_i) + b] \geq 1 - \xi_i \\ & \xi_i \geq 0. \end{aligned} \quad (6)$$

where $C > 0$ is a parameter to be chosen by the user, which corresponds to the strength of the penalty errors, and the ξ_i 's are slack variables that penalize training errors.

Classification of a new data point x is performed by computing the sign of the right side of Equation 5. An important family of kernel functions is the Radial Basis Function, more commonly used for pattern recognition problems [14], which has been used in this paper and is defined by:

$$K(x, y) = e^{-\gamma\|x-y\|^2} \quad (7)$$

where $\gamma > 0$ is a parameter that also is defined by the user.

2.5 Validation and Evaluation of the Classification Methods

In order to validate the classificatory power of the discriminant function, the leave-one-out technique [16] was employed. Through this technique, the candidate nodules from 35 cases in our database were used to train the classifier; the trained classifier was then applied to the candidate nodules in the remaining case. This technique was repeated until all 36 cases in our database had been the “remaining” case.

In order to evaluate the ability of the classifier to differentiate benign from malignant nodules, the area (*AUC*) under the ROC (Receiver Operation Characteristic) [2] curve was used. In other words, the ROC curve describes the ability of the classifiers to correctly differentiate the set of lung nodule candidates into two classes, based on the true-positive fraction (sensitivity) and false-positive fraction (1-specificity). Sensitivity is defined by $TP/(TP + FN)$, specificity is defined by $TN/(TN + FP)$, and accuracy is defined by $(TP + TN)/(TP + TN + FP + FN)$, where TN is true-negative, FN is false-negative, FP is false-positive, and TP is true-positive.

3 Results

The tests described in this paper were carried out using a sample of 36 nodules, 29 benign and 7 malignant. It is important to note that the nodules were diagnosed by physicians and that the diagnosis was confirmed by means of surgery or based on their evolution. Such process takes about two years, which explains the reduced size of our sample.

There were no specific criteria to select the nodules. The sample included nodules with varied sizes and shapes, with homogeneous and heterogeneous characteristics, and in initial and advanced stages of development.

SPSS (*Statistical Package for the Social Sciences*) [17], LIBSVM [18] and NeuralPower [19] were used to training and classification of lung nodules to FLDA, MLP and SVM, respectively. ROCKIT [20] software was used to compute and compare the area under the ROC curve.

Stepwise discriminant analysis [11] was used to select the best variables to differentiate between groups, and the measures selected were NS, VCH and VC (with N as 1). These measures were used to FLDA, MLP and SVM classifiers.

We use the following parameters in the MLP classifier: one hidden layer with four units, hiperbolic tangent as the activation function, the value of 0.15 for the learning ratio, the value of 0.75 for the momentum.

Table 1. Analysis of FLDA, MLP and SVM classifiers

Classifiers	Specificity %	Sensitivity %	Accuracy %	$AUC \pm SE$
FLDA	89.7	71.4	86.1	0.946 ± 0.061
MLP	89.7	85.7	88.8	0.906 ± 0.079
SVM	89.7	57.1	83.3	0.892 ± 0.084

In the classification via SVM a proposed procedure by the authors of LIB-SVM [18] was used to obtain the best constants C and γ with a process of 36-fold cross-validation. In our case, $C = 2.0$ and $\gamma = 2.0$.

Table 1 shows the results of studied classifiers applied to nodule's 3D geometry. Based on the area of the ROC curve, we have observed that: (i) All classifiers have value AUC above 0.800, which means results with accuracy between good and excellent [21]. (ii) SVM have the minor value of sensitivity. (iii) The difference between the ROC curve using the FLDA and the MLP classifiers did not reach statistical significance ($p = 0.641$). The difference between the ROC curve using the FLDA and the SVM classifiers did not reach statistical significance ($p = 0.523$). The difference between the ROC curve using the MLP and the SVM classifiers did not reach statistical significance ($p = 0.799$).

The number of nodules studied in our dataset is too small to allow us to reach definitive conclusions, but preliminary results from this work are very encouraging, demonstrating the potential for multiple variables used in a pattern classification approach to discriminate benign from malignant lung nodules.

4 Conclusion

FLDA, MLP and SVM have been applied to many classifications problems, generally yielding good performance. In this paper, we have compared these three classification algorithms on diagnosis of lung nodule. Results based on the analysis of the ROC curve have shown that the three algorithms had similar performance on this particular task. But a more accurate analysis of the SVM shows that it results in a not so good sensitivity, being less appropriated for a clinical use. Based on these results, we have observed that such measures provide significant support to a more detailed clinical investigation, and the results were very encouraging when nodules were classified with these classifiers. Nevertheless, there is the need to perform tests with a larger database and more complex cases in order to obtain a more precise behavior pattern.

Despite the good results obtained only by analyzing the geometry, further information can be obtained by analyzing the texture. As a future work, we propose a combination of texture and geometry measures for a more precise and reliable diagnosis.

Acknowledgments

We would like to thank Dr. Rodolfo A. Nunes and his team for the clinical support, and the staff from Instituto Fernandes Figueira, particularly Dr. Marcia Cristina Bastos Boechat, for the images provided.

References

1. Tarantino, A.B.: 38. In: *Nódulo Solitário Do Pulmão*. 4 edn. Guanabara Koogan, Rio de Janeiro (1997) 733–753
2. Erkel, A.R.V., Pattynama, P.M.T.: Receiver operating characteristic (ROC) analysis: Basic principles and applications in radiology. *European Journal of Radiology* **27** (1998) 88–94
3. Clunie, D.A.: *DICOM Structured Reporting*. PixelMed Publishing, Pennsylvania (2000)
4. Nikolaidis, N., Pitas, I.: *3-D Image Processing Algorithms*. John Wiley, New York (2001)
5. Silva, A.C., Carvalho, P.C.P.: Sistema de análise de nódulo pulmonar. In: II Workshop de Informática aplicada a Saúde, Itajai, Universidade de Itajai (2002) Available at <http://www.cbcomp.univali.br/pdf/2002/wsp035.pdf>.
6. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*. 3 edn. Addison-Wesley, Reading, MA, USA (1992)
7. da F. Costa, L., Velte, T.J.: Automatic characterization and classification of glangion cells from the salamander retina. *The Journal of Comparative Neurology* **404** (1999) 33–51
8. Zhou, Y., Toga, A.W.: Efficient skeletonization of volumetric objects. *IEEE Transactions on Visualization and Computer Graphics* **5** (1999) 196–208
9. Sonka, M., Hlavac, V., Boyle, R.: *Image Processing, Analysis and Machine Vision*. 2 edn. International Thomson Publishing (1998)
10. Michie, D., Spiegelhalter, D.J., Taylor, C.C.: *Machine Learning, Neural and Statistical Classification*. Ellis Horwood Series in Artificial Intelligence, NJ, USA (1994)
11. Lachenbruch, P.A.: *Discriminant Analysis*. Hafner Press, New York (1975)
12. Duda, R.O., Hart, P.E.: *Pattern Classification and Scene Analysis*. Wiley-Interscience Publication, New York (1973)
13. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford University Press, New York (1999)
14. Haykin, S.: *Redes Neurais: Princípios e Prática*. 2 edn. Bookman, Porto Alegre (2001)
15. Burges, C.J.C.: *A Tutorial on Support Vector Machines for Pattern Recognition*. Kluwer Academic Publishers. (1998)
16. Fukunaga, K.: *Introduction to Statistical Pattern Recognition*. 2 edn. Academic Press, London (1990)
17. Technologies, L.: SPSS 11.0 for windows. Available at <http://www.spss.com> (2003)
18. Chang, C.C., Lin, C.J.: LIBSVM – a library for support vector machines (2003) Available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.
19. Software, C.X.: *Neuralpower professional v. 1.0*. Available at <http://www.geocities.com/neuralpower/> (2003)

20. Metz, C.E.: ROCKIT software. Available at <http://www-radiology.uchicago.edu/krl/toppage11.htm> (2003)
21. Greinera, M., Pfeifferb, D., Smithc, R.: Principles and practical application of the receiver-operating characteristic analysis for diagnostic tests. *Preventive Veterinary Medicine* **45** (2000) 23–41

Diagnosis of Lung Nodule Using Reinforcement Learning and Geometric Measures

Aristófaes Correã Silva¹, Valdeci Ribeiro da Silva Junior²,
Areolino de Almeida Neto¹, and Anselmo Cardoso de Paiva²

¹ Federal University of Maranhão - UFMA, Department of Electrical Engineering,
Av. dos Portugueses, SN, Campus do Bacanga, Bacanga,
65085-580, São Luís, MA, Brazil
`ari@dee.ufma.br`, `areolino@gmx.net`

² Federal University of Maranhão - UFMA, Department of Computer Science,
Av. dos Portugueses, SN, Campus do Bacanga, Bacanga,
65085-580, São Luís, MA, Brazil
`paiva@deinf.ufma.br`, `valdeci_jr@hotmail.com`

Abstract. This paper uses a set of 3D geometric measures with the purpose of characterizing lung nodules as malignant or benign. Based on a sample of 36 nodules, 29 benign and 7 malignant, these measures are analyzed with a technique for classification and analysis called reinforcement learning. We have concluded that this technique allows good discrimination from benign to malignant nodules.

1 Introduction

Lung cancer is known as one of the cancers with shortest survival after diagnosis [1]. Therefore, the sooner it is detected the larger the patient's chance of cure. On the other hand, the more information physicians have available, the more precise the diagnosis will be.

In many cases, it is possible to characterize a nodule as malignant or benign by analyzing its shape. If the nodule is rounded or has a well defined shape, it is probably benign; if it is spiculated or has an irregular shape, it is probably malignant. Figure 1 exemplifies such characteristics. However, in some cases it is hard to distinguish malignant nodules from benign ones [2], [3].

To solve the nodule diagnosis problem we have in general a two phase approach. In the first phase we must extract nodules characteristics that must help us to differentiate the benign from malignant ones. Next phase is devoted to classify based on the extracted characteristics the nodules.

A common approach for the first phase are the model based techniques, that use mathematical to describe the characteristics of lung nodule and therefore how to find them is a set of images.

Statistical and learning based methods are commonly applied to the second phase, such as discriminant analysis, neural network and reinforcement learning [4], [5], [6]. Learning-based methods are a promising techniques, they find









	Benign		Malignant	
Label	(a)	(b)	(c)	(d)
1 Slice				
2 3D Reconstruction				

Fig. 1. Examples of benign lung nodules (a and b) and malignant lung nodules (c and d)

out the characteristics from real images already classified as malignant or benign and “learn” what can define a particular type of nodule.

The purpose of the present work is to investigate the adequacy of the reinforcement learning technique to classify lung nodules based on a set of 3D geometric measures extracted from the lung lesions Computerized Tomography (CT) images.

This paper is organized as follows. Section 2 describes the image database used in the paper, and discusses in detail the geometric measures used to discriminate lung nodules. Tests, discussion and analysis of the application of reinforcement learning to the nodules classification are treated in Section 4. Finally, Section 5 presents some concluding remarks.

2 Material and Methods

2.1 Image Acquisition

The images were acquired with a Helical GE Pro Speed tomography under the following conditions: tube voltage 120 kVp, tube current 100 mA, image size 512×512 pixels, voxel size $0.67 \times 0.67 \times 1.0$ mm. The images were quantized in 12 bits and stored in the DICOM format [7].

2.2 3D Extraction and Reconstruction of Lung Nodule

In most cases, lung nodules are easy to be visually detected by physicians, since their shape and location are different from other lung structures. However, the nodule voxel density is similar to that of other structures, such as blood vessels, which makes automatic computer detection difficult. This happens especially when a nodule is adjacent to the pleura. For these reasons, we have used the 3D region growing algorithm with voxel aggregation [8], which provides physicians greater interactivity and control over the segmentation and determination of required parameters (thresholds, initial slice and seed).

The Marching Cubes algorithm [9] is used to build an explicit representation of volume data. The measures described along the present paper will use this

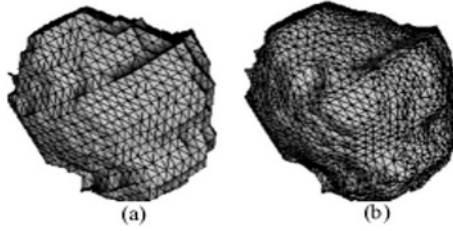


Fig. 2. (a) Application of Marching Cubes. (b) Application of Laplacian technique

representation. In order to remove irregularities of the reconstructed surface, the Laplacian smoothing technique [10] is used. Figures 2 (a) and (b) show the result of applying the Marching Cubes algorithm and the Laplacian technique, respectively.

2.3 3D Geometric Measures

The measures to be presented in this section seek to capture information on the nodule's 3D geometry from the CT. The measures should ideally be invariant to changes in the image's parameters, such as voxel size, orientation, and slice thickness.

Sphericity Index: The Sphericity Index (SI) measures the nodule's behavior in relation to a spherical object. It is defined as

$$SI = \frac{6\sqrt{\pi}V}{A^{\frac{3}{2}}} \quad (1)$$

where V is the surface volume and A corresponds to the surface area. Thus, if the nodule's shape is similar to a sphere, the value will be close to 1. In all cases, $SI \leq 1$.

Convexity Index: The Convexity Index (CI) [11] measures the degree of convexity, defined as the area of the surface of object B ($A(B)$) divided by the area of the surface of its convex hull ($A(H_B)$). That is,

$$CI = \frac{A(B)}{A(H_B)} \quad (2)$$

The more convex the object is, the closer the value of CI will be to 1. For all objects, $CI \geq 1$.

Curvature Index: The two measures presented below are based on the main curvatures k_{min} and k_{max} , defined by

$$k_{\min, \max} = H \mp \sqrt{H^2 - K} \quad (3)$$

where K and H are the Gaussian and mean curvatures, respectively. The values of H and K are estimated using the methods described in [12].

- a) **Intrinsic curvature:** The Intrinsic Curvature Index (*ICI*) [11], [12] captures information on the properties of the surface’s intrinsic curvatures, and is defined as

$$ICI = \frac{1}{4\pi} \int \int |k_{\min} k_{\max}| dA \tag{4}$$

Any undulation or salience on the surface with the shape of half a sphere increments the Intrinsic Curvature Index by 0.5, regardless of its size - that is, the *ICI* counts the number of regions with undulations or saliences on the surface being analyzed.

- b) **Extrinsic curvature:** The Extrinsic Curvature Index (*ECI*) [11], [12] captures information on the properties of the surface’s extrinsic curvatures, and is defined as

$$ECI = \frac{1}{4\pi} \int \int |k_{\max}| (|k_{\max}| - |k_{\min}|) dA \tag{5}$$

Any crack or gap with the shape of half a cylinder increments the *ECI* in proportion to its length, starting at 0.5 if its length is equal to its diameter - that is, the *ECI* counts the number and length (in relation to the diameter) of semicylindrical cracks or gaps on the surface.

Types of Surfaces: With the values of extrinsic (*H*) and intrinsic (*K*) curvatures, it is possible to specify eight basic types of surfaces [13], [14]: *peak* (*K* > 0 and *H* < 0), *pit* (*K* > 0 and *H* > 0), *ridge* (*K* = 0 and *H* < 0), *flat* (*K* = 0 and *H* = 0), *valley* (*K* = 0 and *H* > 0), *saddle valley* (*K* < 0 and *H* > 0), *minimal* (*K* < 0 and *H* = 0), *saddle ridge* (*K* < 0 and *H* < 0).

The measures described below were presented in [15] for the classification of lung nodules and the results were promising. However, they have computed curvatures *H* and *K* directly from the voxel intensity values, while here we compute them in relation to the extracted surface, which is composed by triangles.

In practice, it is difficult to determine values that are exactly equal to zero, due to numerical precision. Therefore we have selected only types *peak*, *pit*, *saddle valley* and *saddle ridge* for our analysis [15].

- a) **Amount of each Surface Type:**

This measure indicates the relative frequency of each type of surface in the nodule, where *APK* (Amount of *peak* surface), *API* (Amount of *pit* surface), *ASR* (Amount of *saddle ridge* surface) and *ASV* (Amount of *saddle valley* surface).

- b) **Area Index for each Surface Type:**

For each surface type, the area occupied in the nodule divided by the total nodule area is computed, where *AIPK* (Area Index for *peak* surface), *AIPI* (Area Index for *pit* surface), *AISR* (Area Index for *saddle ridge* surface) and *AISV* (Area Index for *saddle valley* surface).

- c) **Mean Curvedness for each Surface Type:**

Curvedness is a positive number that measures the curvature amount or intensity on the surface [13]:

$$c = \sqrt{\frac{k_{\min}^2 + k_{\max}^2}{2}} \quad (6)$$

The measures are based on the *curvedness* and the surface types. For each surface type, the mean *curvedness* is determined using the *curvedness* of each surface type, divided by the *curvedness* number in each surface type. Where, *CPK* (mean *curvedness* for *peak*), *CPI* (mean *curvedness* for *pit*), *CSR* (mean *curvedness* for *saddle ridge*) and *CSV* (mean *curvedness* for *saddle valley*).

3 Classification Algorithm

The idea of classification is to encounter a path from the pattern presented to a known target, in the present case to a malignant or to a benign pattern. Furthermore the path found should be the shortest in some sense, in such way that the presented pattern seems to be nearer from a known target and therefore it can be considered of the type of the target. Considering the diverse techniques, the Reinforcement Learning (RL) was chosen to find this path, because it can learn without a mathematical model of a path and because it can learn a correct path only using a reward when the target is encountered [16]. The RL technique presents the following characteristics [17]: Intuitive data; Cumulative learning; and Direct knowledge acquisition.

The first characteristic says that the data manipulated should come from some physical measure or be easily understandable. The second one provides the knowledge to grow up while more data are processed. The last one permits an ease way to store the learning [17].

The RL technique works based on states and actions. The states are a set of variables, each one storing a value of a specific measure. The objective of the states is to configure some situation of an environment. For instance, a geometric figure can be characterized by the number of sides, the size of each side and the angle between each two sides, then for this case there are three different states.

Each set of states has a set of actions, whose objective is to provide a changing in the environment, in order to achieve the desired goal. For example, for a mobile robot in a room, the states can be the position and velocity of an obstacle in this room. The actions are: turn right and turn left. Hence, for a given position and velocity of an obstacle, the turn left action can be decided and for another set of values from these states, to turn right is more adequate, all trying to achieve the goal, in this example to avoid the obstacle [18].

As explained, for each set of state values, there are some actions. Each action has a value, which is obtained during the training phase. This value is used to find the best action, normally the action with the greatest value. As a action can result in a bad or good experience, a (bad or good) reward is given to an action, when it is chosen. Then one can say that a good action taken produces good value for this action and a bad action, a bad value associated. So, a decision taken in the future is based on values, which are obtained in the past, this means that an action will be chosen based on past experiences, which are realized during the

training phase. Therefore, in order to decide an action, the system identifies the actions corresponding to the present set of state values and chooses the action with the greatest value.

But how is obtained the values of each action? The values are obtained during the training. In this phase, many trials are executed to find a path from an initial point to the target. Each successful trial becomes a good reward. This reward is given only to the action chosen, but it can be spread out for the others action executed before. During the training, when an action is chosen, the value associated to this action - $Q(s_t, a_t)$ - is updated considering its present value, the reward obtained with this choice r - and the value of the best action, which will be chosen in the next step - $max_a(Q(s_{t+1}, a_{t+1}))$ -. So the action of the next step can make an influence in the present action, of course in the next trial. The following equation shows the Q-learning updating rule for an action [16].

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r + \gamma max_a(Q(s_{t+1}, a_{t+1})) - Q(s_t, a_t)) \quad (7)$$

The two parameters α and γ are, respectively, the learning rate and the discount rate. The last one is used to increase or not the influences of a previous action in the future actions, in such way that $\gamma = 0$ means one action decided now causes no influences in the next actions. The data structure Q is a matrix, where each cell corresponds to a particular set of states and actions. For example, if the system has three states and two actions, then Q has five dimensions. To be used in a matrix, states and actions must be discretized using any method.

4 Tests and Results

This section shows the results obtained from the application of the RL to a set of lung nodules classification based on its 3D geometric characteristics, discriminating them between malignant from benign.

The tests described in this paper were carried out using a sample of 36 nodules, 29 benign and 7 malignant. It is important to note that the nodules were diagnosed by physicians and had the diagnosis confirmed by means of surgery or based on their evolution. Such process takes about two years, which explains the reduced size of our sample. The sample included nodules with varied sizes and shapes, with homogeneous and heterogeneous characteristics, and in initial and advanced stages of development.

The stepwise analysis [19] selected 5 out of the 13 measures (states), described in Section 2.3, to be analyzed by the reinforcement learning classifier. The selected states were ICE, QPK, QSR, QSV e CPI. Each state was discretized in ten different values. Thus, an action increase, maintain or decrease the present value of the corresponding state, generating five different actions, one for each state. The discretization of each state is shown in Table 1.

With these states and actions, the matrix Q has the following size: $10^5 \times 3^5$, because each state has ten different values and each action three possibilities.

The training was made selecting nineteen images of benign nodules and four malignant, and we choose the more characteristic malignant and benign nodule

Table 1. Discretization of each state

State	ICE	QPK	QSR	QSV	CPI
1	45 - 492.61	9 - 145.88	5 - 167.55	23 - 369.83	0.24 - 0.27
2	492.61 - 1.39e+3	145.89 - 419.67	167.55 - 492.67	369.83 - 1.06e+3	0.27 - 0.32
3	1.39e+3 - 2.29e+3	419.67 - 693.44	492.67 - 817.78	1.06e+3 - 1.76e+3	0.32 - 0.37
4	2.28e+3 - 3.18e+3	693.44 - 967.22	817.77 - 1.14e+3	1.76e+3 - 2.45e+3	0.37 - 0.43
5	3.18e+3 - 4.07e+3	967.22 - 1,240	1.14e+3 - 1,467	2.45e+3 - 3.14e+3	0.43 - 0.48
6	4.07e+3 - 4.97e+3	1,241 - 1.51e+3	1,468 - 1.79e+3	3.14e+3 - 3.84e+3	0.48 - 0.53
7	4.97e+3 - 5.86e+3	1.51e+3 - 1.78e+3	1.79e+3 - 2.12e+3	3.84e+3 - 4.53e+3	0.53 - 0.58
8	5.86e+3 - 6.76e+3	1.79e+3 - 2.06e+3	2.12e+3 - 2.44e+3	4.53e+3 - 5.22e+3	0.58 - 0.64
9	6.76e+3 - 7.65e+3	2.06e+3 - 2.33e+3	2.44e+3 - 2.77e+3	5.22e+3 - 5.92e+3	0.64 - 0.69
10	7.65e+3 - 8,102	2.33e+3 - 2,473	2.77e+3 - 2,931	5.92e+3 - 6,266	0.69 - 0.719

as target. Each image was represented by the five states described above. During the training, each set of states of an image was used as start point of a trip to the target (malignant or benign). Each step of this trip consists of a set of actions, one for each state. One set of actions can be: increase the value of state 1, decrease the value of state 2, and so on. At the end of a trip, when the correct target is found (malignant or benign), the training system provided a reward of value one and zero for others points. After one trial, another trial was made using data of another image. A session training containing all images is an episode. In order to find out the best way, that means, the best action for each set of states, sometimes actions must be chosen randomly. So, in this research, initially the rate of random choice was 50%. The random choice of an action provides a test of another path to the target and posterior comparison with the previous paths.

After the training, the knowledge should have been acquired. This is verified with a test of classification with images not used during the training phase. For this purpose nine benign and two malignant images were selected.

Figure 3 shows the results obtained, where we used the remained nine benign and two malignant nodules. In this figure we represent in the x-axis the nodules case, being cases 1 to 9 benign and cases 10 and 11 malignant. On other hand, the y-axis represent the number of steps from they start point to the target, which means the number of actions taking to reach the case target. When a case take a positive number of steps to reach the target we have a successful classification. Otherwise a negative number represents an incorrect classification and when the classification is not determined we set the number of steps as zero.

The obtained data was generated from four experiments, using 20000, 30000, 40000 and 50000 episodes in the training phase.

The number of right classification grows from 45% for 20000 episodes to 81% for 50000 episodes; as show in Table 2, which indicate a good improvement in the classification success as the number of episodes grows.

An interesting information observed in the results is for 40000 episodes, when the number of successful classification decreased, which should be derived from the random choices used in the training phase, that lead to a poor learning. But, as already proved in RL theorem [16], a very high number of episodes drives to a correct learning, generating a very high successful rate in the classification.

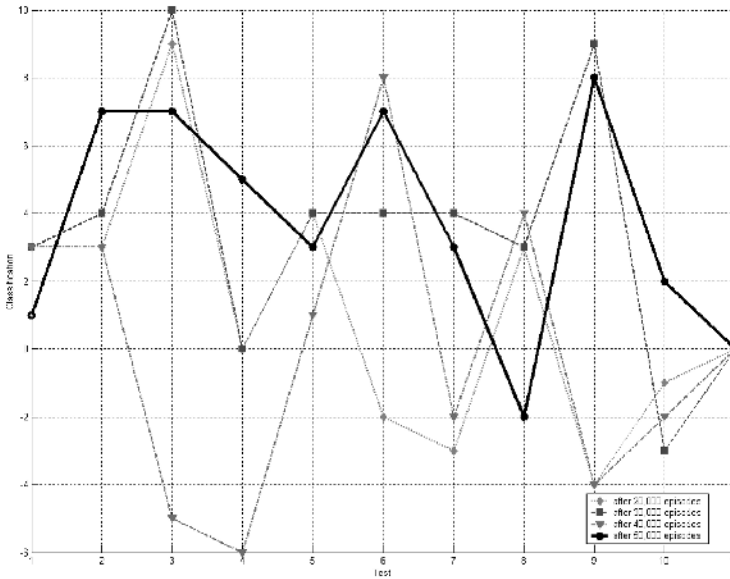


Fig. 3. Application of Reinforcement Learning technique

Table 2. Application of Reinforcement Learning technique

Success	Error	Non-Determined	Number of Episodes
45.45%	36.36%	18.18%	20,000
72.72%	9.09%	18.18%	30,000
45.45%	45.45%	9.09%	40,000
81.81%	9.09%	9.09%	50,000

Due to the relatively small size of the existing CT lung nodule databases and the various CT imaging acquisition protocols, it is difficult to compare the diagnosis performance between the developed algorithms and others proposed in the literature.

5 Conclusion

This paper presented the use of reinforcement learning to solve the problem of lung nodules classification based on 3D geometrics characteristics.

The number of nodules studied in our dataset is too small and the disproportion in the samples does not allow us to make definitive conclusions, However, the results obtained with our sample are very encouraging, demonstrating that the reinforcement learning classifier using characteristics of the nodules' geom-

etry can effectively classify benign from malignant lung nodules based on CT images. Nevertheless, there is the need to perform tests with a larger database and more complex cases in order to obtain a more precise behavior pattern.

Despite the good results obtained we should research to find out a way to shorten the training phase, while maintaining the learning quality. We also must improve our nodules database to generate more definitive results and to make possible the comparison with other classifiers.

Acknowledgments

We thank Dr. Rodolfo Nunes and his team for the clinical support, and Dr. Marcia Boechat with the staff of Instituto Fernandes Figueira, for the images.

References

1. Tarantino, A.B.: 38. In: *Nódulo Solitário Do Pulmão*. 4 edn. Guanabara Koogan, Rio de Janeiro (1997) 733–753
2. Henschke, C.I., et al: Early lung cancer action project: A summary of the findings on baseline screening. *The Oncologist* **6** (2001) 147–152
3. Reeves, A.P., Kostis, W.J.: Computer-aided diagnosis for lung cancer. *Radiologic Clinics of North America* **38** (2000) 497–509
4. Silva, A.C., Carvalho, P.C.P., Gattass, M.: Analysis of spatial variability using geostatistical functions for diagnosis of lung nodule in computerized tomography images. *Pattern Analysis and Applications* **7** (2005) 227–234
5. Silva, A.C., Carvalho, P.C.P., Gattass, M.: Diagnosis of solitary lung nodule using semivariogram and skeletonization in computerized tomography images. 21st Meeting of the Society for Computer Applications in Radiology (SCAR 2004) (2004)
6. Silva, A.C., Carvalho, P.C.P., Peixoto, A., Gattass, M.: Diagnosis of lung nodule using gini coefficient and skeletonization in computerized tomography images, NY, USA, ACM Press New York (2004) 243–248 19th ACM Symposium on Applied Computing (SAC 2004).
7. Clunie, D.A.: *DICOM Structured Reporting*. PixelMed Publishing, Pennsylvania (2000)
8. Nikolaidis, N., Pitas, I.: *3-D Image Processing Algorithms*. John Wiley, New York (2001)
9. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics* **21** (1987) 163–169
10. Ohtake, Y., Belyaev, A., Pasko, A.: Dynamic meshes for accurate polygonization of implicit surfaces with shape features. In Press, I.C.S., ed.: *SMI 2001 International Conference on Shape Modeling and Applications*. (2001) 74–81
11. Smith, A.C.: *The Folding of the Human Brain, from Shape to Function*. PhD thesis, University of London (1999) Available at <http://carmen.umds.ac.uk/a.d.smith/phd.html>.
12. Essen, D.C.V., Drury, H.A.: Structural and functional analyses of human cerebral cortex using a surface-based atlas. *The Journal of Neuroscience* **17** (1997) 7079–7102
13. Koenderink, J.J.: *Solid Shape*. MIT Press, Cambridge, MA, USA (1990)

14. Henderson, D.W.: *Differential Geometry: A Geometric Introduction*. Prentice-Hall, Upper Saddle River, New Jersey (1998)
15. Kawata, Y., Niki, N., Ohmatsu, H., Kakinuma, R., Eguchi, K., Kaneko, M., Moriyama, N.: Classification of pulmonary nodules in thin-section CT images based on shape characterization. In: *International Conference on Image Processing*. Volume 3., IEEE Computer Society Press (1997) 528–530
16. Barto, A., Sutton, R., Anderson, C.: Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man and Cybernetics* **13** (1983) 834–846
17. Almeida, A., Heimann, B., Góes, L., Nascimento, C.: Obstacle avoidance in dynamic environment: A hierarchical solution. Volume 6., São Paulo, Congresso Brasileiro de Redes Neurais (2003) 289–294
18. Almeida, A., Heimann, B., Góes, L., Nascimento, C.: Avoidance of multiple dynamic obstacles. In: *International Congress of Mechanical Engineering*. Volume 17., São Paulo (2003)
19. Duda, R.O., Hart, P.E.: *Pattern Classification and Scene Analysis*. Wiley-Interscience Publication, New York (1973)

Iris Recognition Algorithm Based on Point Covering of High-Dimensional Space and Neural Network

Wenming Cao^{1,2}, Jianhui Hu¹, Gang Xiao¹, and Shoujue Wang²

¹The College of Information Engineering, ZheJiang University of Technology,
Hangzhou, 310014, China

²Lab of Artificial Neural Networks, Institute of Semiconductors,
CAS, Beijing, 100083, China
csann@zjut.edu.cn

Abstract. In this paper, we constructed a Iris recognition algorithm based on point covering of high-dimensional space and Multi-weighted neuron of point covering of high-dimensional space, and proposed a new method for iris recognition based on point covering theory of high-dimensional space. In this method, irises are trained as “cognition” one class by one class, and it doesn't influence the original recognition knowledge for samples of the new added class. The results of experiments show the rejection rate is 98.9%, the correct cognition rate and the error rate are 95.71% and 3.5% respectively. The experimental results demonstrate that the rejection rate of test samples excluded in the training samples class is very high. It proves the proposed method for iris recognition is effective.

1 Introduction

In recent years, with the development of information technology and the increasing need for security, intelligent personal identification has become a very important and urgent problem. The emerging biometric technology can solve the problem, which takes the unique, reliable and stable biometric features (such as fingerprints, iris, face, palm-prints, gait etc.) as identification body. This technology has very high security, reliability and effectivity. As one of the biometric technology, iris recognition has very high reliability. Comparing with other biometric identification technology, the fault acceptance rate and the fault rejection rate of iris recognition are very low. The technology of iris recognition has many advantages, i.e., stability, non-invasiveness, uniqueness. All these desirable properties make the technology of iris recognition have very high commercial value. Based on the above reasons, many researchers have applied themselves to this field. Daugman used multi-scale quadrature wavelets to extract texture-phase structure information of iris to generate a 2048-bit iriscode and compared the difference between a pair of iris representations by computing their Hamming distance via the XOR operator [1],[2]. Wildes et al. represented the iris texture with a Laplacian pyramid constructed with four different resolution levels and used the normalized correlation to determine whether the input image and the model

image are from the same class [3]. Boles et al. calculated zero-crossing representation of 1D wavelet transform at various resolution levels of a virtual circle on an iris image to characterize the texture of the iris. Iris mating was based on two dissimilarity functions [4][10][11]. In this paper, from the cognition science point of view, we constructed a neuron of point covering of high-dimensional space[5][6][7], and propose a new method for iris recognition based on point covering theory of high-dimensional space and neural network[8][12]. The results of experiments show the rejection rate is 98.9%, the correct recognition rate and the error rate are 95.71% and 3.5% respectively. The experimental results demonstrate that the rejection rate of test samples excluded in the training samples class is very high. It proves the proposed method for iris recognition is effective.

The remainder of this paper is organized as follows. Section 2 describes image preprocessing. Section 3 introduces iris recognition algorithm based on point covering theory of multi-dimensional space and neural network. Experiments results and experimental analysis are given in Section 4 and Section 5 respectively.

2 Image Preprocessing

Iris image preprocessing is mainly composed of iris localization, iris normalization and enhancement.

2.1 Iris Localization

Iris localization namely is the localization of the inner boundary and the outer boundary of a typical iris can approximately be taken as circles. It is the important part of the system of iris recognition, and exact localization is the premise of the iris identification and verification.

2.1.1 Localization of the Inner Boundary

The original iris image (see Fig.1(a)) has some character of the gray-scale distribution. The iris is darker than the sclera, and the pupil is greatly darker than the iris, as shown in Fig.1(a). From the histogram (see Fig.1(b)), we can clearly see that the low gray-scale mainly converges at the first peak value. Therefore, we adopt the binary transform to localize the inner boundary. From the image after binary transform (see Fig.2(a)), we find that the areas of zero gray-scale are almost the areas of the pupil and eyelash. Therefore, we reduce the influence of the eyelash by erode and dilation (see Fig.2(a)).

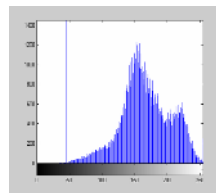
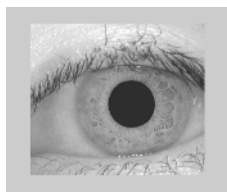


Fig. 1. (a) original image

(b) histogram of the iris

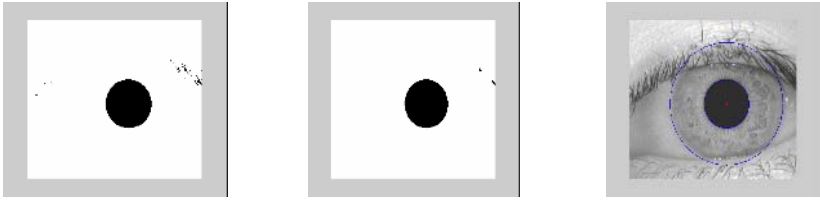


Fig. 2. (a) binary image (b) binary image after erode and dilation (c) localized image

From the Fig 2(b), we can find that the length and the midpoint of the longest chord can be taken as the approximate diameter and center of the pupil respectively. Namely, according the geometry knowledge, let the length of the longest chord is dia_{max} , and the coordinates of the first point of the chord are $xbegin$ and $ybegin$, then

$$xpupil = xbegin + \frac{dia_{max}}{2}, ypupil = ybegin, rpupil = \frac{dia_{max}}{2} \quad (1)$$

Where $xpupil$ and $ypupil$ denote the center coordinates of the pupil, and $rpupil$ denotes the radius of the pupil.

When the quality of the image is reliable, this algorithm can localize the pupil quickly and exactly. Otherwise, we can correct the method as follow:

1. We can reduce the searching area by subtracting the pixels on the edge of the image.
2. We can get k chords, which are less than a certain threshold near the longest chord, and take the average value of center coordinates of k chord as the center of the pupil.

2.1.2 Localization of the Outer Boundary

The exact parameters of the outer boundary are obtained by using edge detection (Canny operator in our experiments) and Hough transform. The image after Edge detection includes some useless points. For eliminating the influence, we remove the useless points between the areas of $[30^\circ, 150^\circ]$ and $[225^\circ, 315^\circ]$ according to the center of the pupil. Then, Hough transform is adopted to localize the outer boundary.

By the above method, we can localize the inner boundary and the outer boundary of the iris exactly. The localizations results of the iris are showed in Fig.2(c).

2.2 Iris Normalization and Enhancement

Iris from different people may be captured in different size, and even for irises from the same eye, the size may change because of illumination variations and other factors (the pupil is very sensitive to lighting changes). Such elastic deformation in iris texture will influence the results of iris recognition. For the purpose of achieving more accurate recognition results, it is necessary to compensate for such deformation.

In our experiment, every point of the iris image is mapped to the polar coordinates by the following formula.

$$\begin{cases} x(r, \theta) = (1 - r)x_p(\theta) + rx_s(\theta) \\ y(r, \theta) = (1 - r)y_p(\theta) + ry_s(\theta) \end{cases} \quad (2)$$

In which, $(x_p(\theta), y_p(\theta))$ and $(x_s(\theta), y_s(\theta))$ denote the point of intersection with the inner boundary and the outer boundary respectively.

In our experiment, the sector areas $([130^\circ, 230^\circ])$ and $([310^\circ, 410^\circ])$ are intercepted for normalization according the pupil center. In this way, one hand, it is simple; on the other hand, the segment texture information is enough to identify the different persons. Then, the iris ring is unwrapped to a rectangular texture block with a fixed size (64×256) , and the rows correspond to the radius and the columns correspond to the angles (see Fig.3(a)). The normalized iris image still has low contrast and may have non-uniform brightness caused by the position of light sources. All these may affect the feature analysis. Therefore, we enhance the normalized image by means of histogram equalization. Such processing compensates for non-uniform illumination, as well as improving the contrast of the image. The enhanced image is shown in Fig.3(b).

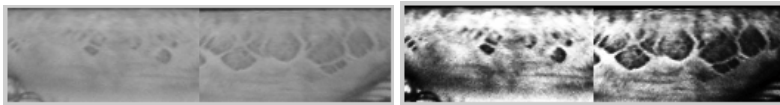


Fig. 3. (a) normalized image (b)enhanced image

3 Iris Recognition Algorithm Based on Point Covering of Multi-dimensional Space and Neural Network

Multi-weighted neuron can be represented as following formula:

$$Y = f[\Phi(X, W_1, W_2, \dots, W_m) - Th] \quad (3)$$

In which, $\Phi(X, W_1, W_2, \dots, W_m)$ denotes the relation between the input point X and m weight (W_1, W_2, \dots, W_m) . Let $m = 3$, it is 3-weighted neuron, named *pSi3*. And *pSi3* can be described as follow:

$$Y = f[\Phi(X, W_1, W_2, W_3) - Th] \quad (4)$$

$$\Phi(X, W_1, W_2, W_3) = \|X - \theta_{(w_1, w_2, w_3)}\| \quad (5)$$

In which, $\theta_{(W_1, W_2, W_3)}$ denotes the finite area, which is enclosed by three points (W_1, W_2, W_3) , and it is a triangle area. Namely, $\theta_{(W_1, W_2, W_3)}$ can be represented as follow:

$$\theta_{(W_1, W_2, W_3)} = \{Y | Y = \alpha_2 [\alpha_1 W_1 + (1 - \alpha_1) W_2] + (1 - \alpha_2) W_3, \alpha_1 \in [0, 1], \alpha_2 \in [0, 1]\} \quad (6)$$

Then, $\Phi(X, W_1, W_2, W_3) - Th$ actually is the Euclid distance from X to the triangle area of the $pSi3$ neuron. The model of activation function is:

$$f(x) = \begin{cases} 1, & x \leq Th \\ -1, & x > Th \end{cases} \quad (7)$$

In multi-dimensional space, we use every three sample's points of the same class to construct a finite 2D plane, namely, a triangle. Then several 2D spaces can be constructed, and we cover these planes by the $pSi3$ neuron to approximate the complicated "shape", which is formed by many sample points of the iris in multi-dimensional space.

3.1 Construction of Point Covering Area of Multi-dimensional Space

Step 1: Let the sample points of the training set are $\alpha = \{A_1, A_2, \dots, A_N\}$. In which, N is the number of the total sample points. To figure out the distance of every two points, the two points having the least distance are defined as B_{11} and B_{12} . Let B_{13} denotes the nearest point away from B_{11} and B_{12} , and B_{13} must doesn't in the line formed by B_{11} and B_{12} . In this way, B_{11} , B_{12} and B_{13} construct the first triangle plane represented as θ_1 , which is covered by a $pSi3$ neuron, the covering area is:

$$P_1 = \{X | \rho_{X\theta_1} \leq Th, X \in R^n\} \quad (8)$$

$$\theta_1 = \{Y | Y = \alpha_2 [\alpha_1 B_{11} + (1 - \alpha_1) B_{12}] + (1 - \alpha_2) B_{13}, \alpha_1 \in [0, 1], \alpha_2 \in [0, 1]\} \quad (9)$$

Where $\rho_{X\theta_1}$ denotes the distance from X to θ_1 .

Step 2: Firstly, The rest points contained in P_1 should be removed. Then, according to the method of step1, define the nearest point away from B_{11} , B_{12} and B_{13} as B_{21} . Among B_{11} , B_{12} and B_{13} , two nearest points away from B_{21} are denoted as B_{22} and B_{23} . And B_{21} , B_{22} and B_{23} construct the second triangle defined as θ_2 , which is covered by another $pSi3$ neuron. And the covering area is described as follow:

$$P_2 = \{X \mid \rho_{X\theta_2} \leq Th, X \in R^n\} \tag{10}$$

$$\theta_2 = \{Y \mid Y = \alpha_2[\alpha_1 B_{21} + (1 - \alpha_1)B_{22}] + (1 - \alpha_2)B_{23}, \alpha_1 \in [0,1], \alpha_2 \in [0,1]\} \tag{11}$$

Where $\rho_{X\theta_2}$ denotes the distance from X to θ_2 .

Step 3: Remove the rest points contained in the covering area of the front $(i - 1)$ $pSi3$ neurons. Let B_{i1} denotes the nearest point from the remained points to the three vertexes of the $(i - 1)th$ triangle. Two nearest vertexes of the $(i - 1)$ triangle away from B_{i1} are represented as B_{i2} and B_{i3} . Then, B_{i1} , B_{i2} and B_{i3} construct the ith triangle, defined as θ_3 .

In the same way, θ_3 is covered by a $pSi3$ neuron. The covering area is

$$P_i = \{X \mid \rho_{X\theta_2} \leq Th, X \in R^n\} \tag{12}$$

$$\theta_3 = \{Y \mid Y = \alpha_2[\alpha_1 B_{i1} + (1 - \alpha_1)B_{i2}] + (1 - \alpha_2)B_{i3}, \alpha_1 \in [0,1], \alpha_2 \in [0,1]\} \tag{13}$$

Step 4: Repeat the step 3 until all sample points are conducted successfully. Finally, there are m $pSi3$ neurons, and their mergence about covering area is the covering area of every iris' class.

$$P = \bigcup_{i=1}^m P_i \tag{14}$$

3.2 Iris Recognition Algorithm Based on Point Covering of High-dimensional Space

Taking $Th = 0$ under recognition, the $pSi3$ neuron can be described as follow:

$$\rho = \|X - \theta_{(w_1, w_2, w_3)}\| \tag{15}$$

The output ρ is the distance from X to the finite area $\theta_{(w_1, w_2, w_3)}$.

The distance from X to the covering area of the ith class iris is:

$$\rho_i = \min_{j=1}^{M_i} \rho_{ij}, \quad i = 1, \dots, 80 \tag{16}$$

In which, M_i denotes the number of the $pSi3$ neuron of the ith iris, ρ is the distance from X to the covering area of the jth neuron of the ith class' iris.

The X will be classified to the iris class corresponding to the least ρ_i . Namely, the classification method is:

$$j = \min_{i=1}^{80} \rho_i, \quad j \in (1, \dots, 80) \quad (17)$$

4 Experimental Results

Images of CASIA (Institute of Automation, Chinese Academy of Sciences) iris image database are used in this paper. The database includes 742 iris images from 106 different eyes (hence 106 different classes) of 80 subjects. For each iris class, images are captured in two different sessions and the interval between two sessions is one month. The experiment processes and experiment results are presented as follow:

(1) In our experiment, 3 random samples from each class in the frontal 80 classes (hence, 240 samples) are chosen for training, and a $pSi3$ neuron of multi-weighted neural network is constructed for the 3 samples. Five samples from the training set are shown in Fig.4. Then, the entire iris database is taken as test sample set. In which, 182

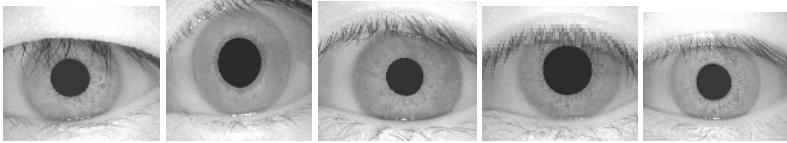


Fig. 4. Iris samples from the training set

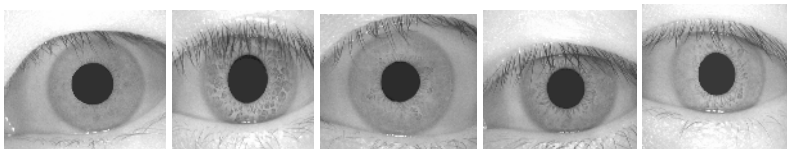


Fig. 5. Iris samples from the second test set

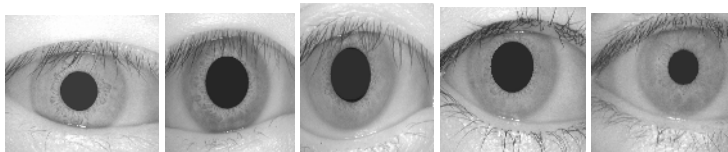


Fig. 6. Iris samples from the first test set

(26×7) samples, which don't belong to the classes of training samples, are referred to the first sample set. The remainder of total 560 (80×7) samples is referred to the second sample set. Fig.5 shows five samples from the second test set and Fig.6 shows five samples from the first test set.

(2) The rejection rate=the number of samples which are rejected correctly in the first sample set/the total number of the first sample set. The correct cognition rate=the number of samples which are recognized corrected in the second sample set / the total number of the second sample set. The error recognition rate=(the number of samples which are recognized mistakenly in the first sample set +the number of samples which are recognized mistakenly in the second sample set) / the total number of the second sample set.

(3) For total 742 test samples, 180 samples are rejected correctly and the other 2 samples are recognized mistakenly in the first test sample; and 536 samples are recognized correctly and the rest 24 samples are recognized mistakenly in the second test sample. Therefore, the rejection rate is 98.9%(180/182), the correct cognition rate and the error recognition rate are 95.71%(536/560) and 3.5%((2+24)/742) respectively.

5 Experimental Analysis

We can conclude from the above experimental results that:

(1) Irises are trained as "cognition" one class by one class in our method, and it doesn't influence the original recognition knowledge for samples of the new added class.

(2) Although the correct cognition rate is not very well, the result of rejection is wonderful. In our experiment, the rejection rate is 98.9%, namely, the iris classes that don't belong to the training test can be rejected successfully.

(3) The iris recognition algorithm based on neuron of multi-weighted neural network is applied in the experiment and the total samples of every class construct the shape of 1D distribution. Namely, it is the network connection of different neuron.

(4) The distribution of the recognized thing should be researched firstly when we apply the algorithm for iris recognition based on point covering theory of high-dimensional space. Then, the covering method of neural network is considered.

(5) In above experiment, if the image preprocessing is more perfectly, the experimental results maybe better.

To sum up, it proves the proposed iris recognition algorithm based on point covering of high-dimensional space and neural network is effective.

References

- [1] J.Daugman, Biometric Personal Identification System Based on Iris Analysis [P]. U.S. Patent 5291560, 1994.
- [2] J.Daugman, High Confidence Visual Recognition of Persons by a Text of Statistical Independence, IEEE Trans. Pattern Anal. Mach. Intell. 15(11) (1993) 1148-1161.

- [3] R.Wildes, Iris Recognition: An Emerging Biometric Technology, Proc. IEEE 85(1997) 1348-1363.
- [4] W.Boles, B. Boashash, A Human Identification Technique Using Image of the Iris and Wavelet Transform, IEEE Trans. Signal Process. 46(4) (1998) 1185-1188.
- [5] Wang Shoujue, Li Zhaozhou, Chen Xiangdong, Wang Bainan, Discussion on the Basic Mathematical Models of Neurons in General Purpose Neurocomputer, ACTA ELECTRONICA SINICA. 2001, 29(5): 577-580
- [6] Wang Shoujue, Xu Jian, Wang Xianbao, Qin Hong, Multi-camera Human-face Personal Identifications System Based on the Bionic Pattern Recognition, ACTA ELECTRONICA SINICA. 2003, 31(1): 1-3
- [7] Wang ShouJue, A New Development on ANN in China - Biomimetic Pattern Recognition and Multi weight Vector Neurons, LECTURE NOTES IN ARTIFICIAL INTELLIGENCE 2639: 35-43 2003
- [8] Yang Wen, Yu Li, et al, A Fast Iris Location Algorithm, Computer Engineering and Applications, 2004.10
- [9] Wang Yunhong, Zhu Yong, Tan Tieniu, Biometrics Personal Identification Based on Iris Pattern[J], ACTA AUTOMATICA SINICA 2002. 28(1):1-10.
- [10] Li Ma et al. Local Intensity Variation Analysis for Iris Recognition, Pattern Recognition 37(2004) 1287-1298.
- [11] Han Fang, Chen Ying, Lu Hengli, An Effective Iris Location Algorithm, Journal of Shanghai University (Natural Science). 2001.7(6):01-03.
- [12] Wenming Cao, Feng Hao, Shoujue Wang: The application of DBF neural networks for object recognition. Inf. Sci. 160(1-4): 153-160 (2004)

Automatic Clinical Image Segmentation Using Pathological Modelling, PCA and SVM

Shuo Li¹, Thomas Fevens¹, Adam Krzyżak¹, and Song Li²

¹ Medical Imaging Group,
Department of Computer Science and Software Engineering,
Concordia University, Montréal, Québec, Canada
{shuo_li, fevens, krzyzak}@cs.concordia.ca

² School of Stomatology, Anhui Medical University,
Hefei, Anhui, P.R. China
xlisong@sohu.com

Abstract. A general automatic method for clinical image segmentation is proposed. Tailored for the clinical environment, the proposed segmentation method consists of two stages: a learning stage and a clinical segmentation stage. During the learning stage, manually chosen representative images are segmented using a variational level set method driven by a pathologically modelled energy functional. Then a window-based feature extraction is applied to the segmented images. Principal component analysis (PCA) is applied to these extracted features and the results are used to train a support vector machine (SVM) classifier. During the clinical segmentation stage, the input clinical images are classified with the trained SVM. By the proposed method, we take the strengths of both machine learning and variational level set while limiting their weaknesses to achieve automatic and fast clinical segmentation. Both chest (thoracic) computed tomography (CT) scans (2D and 3D) and dental X-rays are used to test the proposed method. Promising results are demonstrated and analyzed. The proposed method can be used during preprocessing for automatic computer aided diagnosis.

Keywords: Image segmentation, support vector machine, machine learning, principal component analysis, dental X-rays.

1 Introduction

Image segmentation is an important component of medical imagery which plays a key role in computer assisted medical diagnosis. Segmentation of medical images is typically more challenging than the segmentation of images in other fields. This is primarily due to a large variability in topologies, the complexity of medical structures and poor image modalities such as noise, low contrast, several kinds of artifacts and restrictive scanning methods. This is especially true for volumetric medical images where a large amount of data is coupled with complicated 3D anatomical structures. This paper reports innovative work

using machine learning techniques such as the support vector machine (SVM) and principal component analysis (PCA) learning with a pathologically modelled variational level set method to address the most challenging problems in the medical image analysis: clinical image segmentation and analysis. Although the SVM has been used in image segmentation, it is usually used during an intermediate step [1, 2, 3]. This is the first such work which uses the SVM directly for medical image segmentation, to the best of our knowledge.

One of latest techniques in medical image segmentation is based on a class of deformable models, referred as “level set” or “geodesic active contours/surfaces.” The application of the level set method in medical image segmentation is extremely popular due to its ability to capture the topology of shapes in medical imagery. Codimension-two geodesic active contours were used in [4] for tubular structures. The fast marching algorithm [5] and level set method were used in [6] and [7], while Region competition, introduced in [8], was used in [9]. In [2, 3, 10], Li *et al.* applied a variational level set segmentation approach, accelerated by an SVM, for medical image segmentation, analysis and visualization.

Although efficient, level set methods are not suitable for general use clinical image segmentation due to several reasons: (1) high computational cost; (2) complicated parameter settings; (3) sensitivity to the placement of initial contours. With regard to the latter, as will be shown in experimental results, the running time of the level set method heavily relies on the position and size of initial curves and geometric and topological complexity of objects. Moreover for some cases, the coupled level set method does not converge for some initial curves.

To overcome the current challenges in clinical image segmentation, in this paper, we combine the level set method approach with a machine learning technique. We employ the level set method only during the training stage of the SVM which limits the effect of the weaknesses (i.e., the slowness and lack of stability) of the level set method. Through the application of PCA, we then use the SVM exclusively for segmentation which leads to faster and more robust segmentation.

2 Proposed Method

The proposed method consists of two stages: a learning stage and a clinical segmentation stage. During the segmentation stage, a variational level set method driven by a pathologically modelled energy functional is used. This is followed by window-based feature extraction using PCA analysis. The extracted features are used to train an SVM. During the clinical segmentation, the clinical image is directly segmented by the trained SVM.

2.1 Level Set Method

Proposed by Osher and J. Sethian [5], level set methods have attracted much attention from researchers from different areas. In problems of curve evolution, the level set method and in particular the motion by mean curvature of Osher and Sethian [5] have been used extensively. This is because these methods allow for

curve characteristics such as cusps, corners, and automatic topological changes. Moreover, the discretization of the problem is made on the regular grid.

Let Ω be a bounded open subset of R^2 , with $\partial\Omega$ as its boundary. Let $U_0: \Omega \rightarrow R$ be a given image, and $C : [0, 1] \rightarrow R^2$ be a parameterized curve. The curve C is represented implicitly via a Lipschitz function ϕ , where $C = \{(x, y) | \phi(x, y) = 0\}$, and the evolution of the curve is given by the zero-level curve at time t as the function $\phi(t, x, y)$. Evolving the curve C in normal direction with speed F leads to the differential equation

$$\begin{cases} \frac{\partial\phi}{\partial t} = |\nabla\phi|F \\ \phi(0, x, y) = \phi_0(x, y) \end{cases} \tag{1}$$

where the set $C = \{(x, y) | \phi_0(x, y) = 0\}$ defines the initial contour. A particular case is the motion by mean curvature, when $F = \text{div}(\frac{\nabla\phi}{|\nabla\phi|})$ is the curvature.

2.2 Variational Level Set Method

Chan *et al.* [11, 12] proposed an Mumford-Shah functional for level set segmentation. They add a minimal variance term E_{MV} . The model is able to detect contours both with or without a gradient. Objects with smooth boundaries or even with discontinuous boundaries can be successfully detected. Moreover they claim this model is robust to the position of initial the initial contour. The 2D version of the model can be expressed as

$$\inf_{(c_1, c_2, C)} E = \mu \cdot \text{Length}(C) + v \cdot \text{Area}(\text{Inside}(C)) + E_{MV}.$$

with

$$E_{MV} = \lambda_1 \int_{\text{inside}(C)} |u_0(x, y) - c_1| dx dy + \lambda_2 \int_{\text{outside}(C)} |u_0(x, y) - c_2| dx dy$$

where c_i are the averages of u_0 inside and outside C , and $\mu \geq 0, v \geq 0, \lambda_1 > 0$ and $\lambda_2 > 0$ are fixed parameters.

The level set function they obtain is given by

$$\begin{cases} \frac{\partial\phi}{\partial t} = \delta_\varepsilon(\phi) [\mu \cdot \text{div}(\frac{\nabla\phi}{|\nabla\phi|}) - v - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2] = 0 \\ \phi(0, x, y) = \phi_0(x, y) \text{ in } \Omega \\ \frac{\delta_\varepsilon(\phi)\partial\phi}{|\nabla\phi|\partial\mathbf{n}} = 0 \text{ on } \partial\Omega. \end{cases}$$

where \mathbf{n} denotes the exterior to the boundary $\partial\Omega$, $\frac{\partial\phi}{\partial\mathbf{n}}$ denotes the normal derivative of ϕ at the boundary and δ_ε is the Dirac delta function.

The Chan and Vese functional is very good for segmenting an image into two regions. To segment images with multiple regions we use Samson’s method. In [13], Samson *et al.* presented a variational approach as shown in Eqs. 2 and 3.

$$\begin{aligned} \inf E = & \sum_{1 \leq i \leq j \leq n} f_{ij} \text{Length}(\Gamma_{ij}) + \sum_{1 \leq i \leq n} v_i \text{Area}(\text{Inside}(C_i)) \\ & + \sum_i \int_{\Omega_i} e_i \frac{(u_0 - c_i)^2}{\sigma_i^2} dx dy + \frac{\lambda}{2} \int (\sum_{j=1}^n H(\phi_j) - 1)^2 dx dy. \end{aligned} \tag{2}$$

where Γ_{ij} is the intersection of different regions and σ_i is the variance. The level set function they obtain is given by

$$\begin{cases} \frac{\partial \phi_i}{\partial t} = \delta_\varepsilon(\phi_i) \left(\gamma_i \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - e_i \frac{(u_0 - c_i)^2}{\sigma_i^2} - \lambda \left(\sum_{j=1}^n H(\phi_j) - 1 \right) \right) \\ \frac{\partial \phi_i}{\partial \mathbf{n}} = 0 \text{ on } \partial \Omega. \end{cases} \quad (3)$$

where $H(\cdot)$ is the Heaviside function.

2.3 Pathologically Modelled Variational Level Set Method

In this work, we apply the variational level set method to segment the representative images. First, with the assistance of a doctor or clinician, the energy functional will be modelled according to the pathological meaning of different regions in an image. In the following we are going to take chest CT (2D and 3D) scans and dental X-ray images as examples as can be seen in Fig. 1.

Chest CT Scan. Fig. 1(a) demonstrates a pathological modelling for chest (thoracic) computed tomography (CT) scans. The images can be divided into four regions of interest: the Background Region (Ω_{BR}), the Skeletal Structure (bone) Region (Ω_{SR}), the Fatty Tissue Region (Ω_{FR}) and the Muscle and Visceral Tissue Region (Ω_{MR}). Energy functional for the four coupled level set functions are modelled as Eq. 4.

$$E_{MV}(\phi_i) = \int_{\Omega_{BR}} \frac{e_1(u - c_{BR})^2}{\sigma_{NR}^2} dx dy + \int_{\Omega_{FR}} \frac{e_2(u - c_{FR})^2}{\sigma_{FR}^2} dx dy + \int_{\Omega_{SR}} \frac{e_3(u - c_{SR})^2}{\sigma_{SR}^2} dx dy + \int_{\Omega_{MR}} \frac{e_4(u - c_{MR})^2}{\sigma_{MR}^2} dx dy \quad (4)$$

where $c_i, i=1, \dots, 4$, is the mean grey value of region Ω_i .

Dental X-ray. With prior information, this pathological modelling can also be used for computer aided diagnosis. As shown in the Figs. 1 (b) and (c), X-ray images can be divided into four regions of interest: the Normal Region (Ω_{NR}), the Potentially Abnormal Region (Ω_{PAR}), the Abnormal Region (Ω_{AR}) and the Background Region (Ω_{BR}). Since Ω_{AR} and Ω_{BR} are not separable in terms of intensity values, so in the segmentation, we take Ω_{AR} and Ω_{BR} to be one

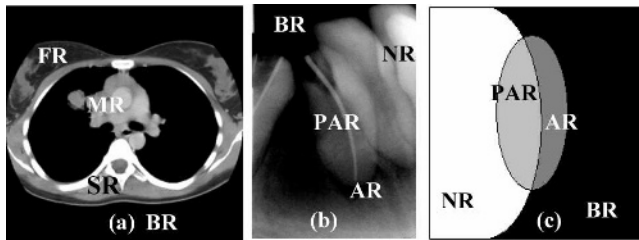


Fig. 1. Pathological modelling for chest CT scans (a) and dental X-rays (b and c)

region: the Abnormal and Background Region (Ω_{ABR}). Energy functional for three coupled level set functions are modelled as Eq. 5.

$$E_{MV}(\phi_i) = e_1 \int_{\Omega_{NR}} \frac{(u - c_{NR})^2}{\sigma_{NR}^2} dx dy + e_2 \int_{\Omega_{PAR}} \frac{(u - c_{PAR})^2}{\sigma_{PAR}^2} dx dy + e_3 \int_{\Omega_{ABR}} \frac{(u - c_{ABR})^2}{\sigma_{ABR}^2} dx dy. \tag{5}$$

The proposed pathological modelling explicitly incorporates regions of problems as part of the modelling, the identification of such areas would be an automatic byproduct of the segmentation. Moreover those problem regions generally indicate some possible areas of bone loss in teeth or the jaw or root decay, which are the primary reasons that X-rays are taken in many countries. Early detection of bone loss and root decay is very important since often they can be remedied by dental procedures, such as a root canal, for example. Without early treatment, bone loss may lead to tooth loss or erosion of the jaw bone.

2.4 Learning

As shown in Fig. 2, the learning phase consists of several steps. First, manually chosen images are segmented by the variational level set described in section 2.3. To avoid distraction, the high uncertainty areas are removed. Next, window-based feature extraction is applied. The results will be used to train the SVM after applying PCA learning to extract features.

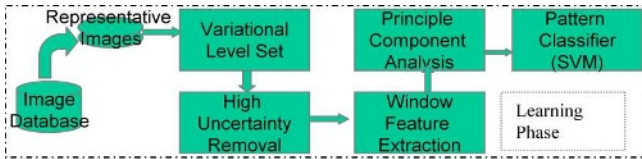


Fig. 2. Learning phase diagram

Uncertainty Removal. Before feature extraction, those areas of high uncertainty in the segmented image will be removed to avoid the possible distraction. The uncertainty measurement is the product of two components: a numerical solution uncertainty component $\psi_1(x, y)$ and a variance uncertainty component $\psi_2(x, y)$ as shown:

$$\psi(x, y) = \psi_1(x, y) \cdot \psi_2(x, y) = \frac{1 + \max(H(\phi_i))}{1 + \sum H(\phi_i)} \cdot \frac{\sum \sigma_i H(\phi_i)}{\sum |u - c_i| H(\phi_i)}$$

Feature Extraction and Principal Component Analysis. A window-based feature extraction is applied to each segmented region in the image. This is

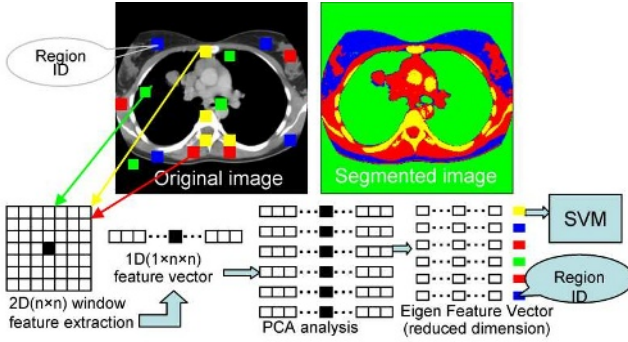


Fig. 3. Feature extraction diagram

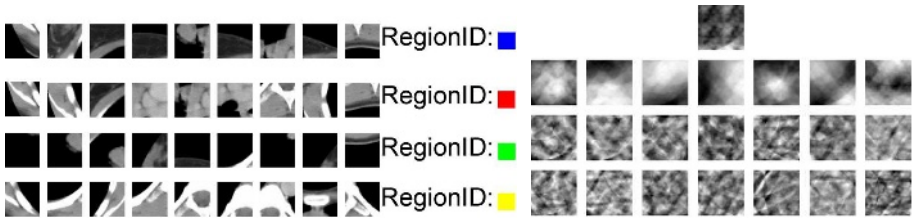


Fig. 4. Window based features

Fig. 5. Average patch (first row) and eigen patches

illustrated in Fig. 4. The PCA method used here is adapted from [14, 15]. Let the features Γ_i ($i = 1..M$) constitute the training set (Γ). The average matrix ($\bar{\Gamma}$) and covariance matrix C are:

$$\begin{aligned}
 \bar{\Gamma} &= \frac{1}{M} \sum_{i=1}^M \Gamma_i \\
 \Phi_i &= \Gamma_i - \bar{\Gamma} \\
 C &= \frac{1}{M} \sum_{i=1}^M \Phi_i^T \Phi_i = AA^T \\
 L &= A^T A (L_{n,m} = \Phi_n^T \Phi_m) \\
 u_i &= \sum_{k=1}^M v_{ik} \Phi_k (l = 1, \dots, M)
 \end{aligned} \tag{6}$$

where L is a $M \times M$ matrix, v_{ik} are the M eigenvectors of L and u_i are eigen-patches which was called eigenfaces in [14, 15]. The advantage of the PCA analysis here is its ability to remove the effects of noise and also to accelerate the classification by reduced feature dimension.

SVM Training and Segmentation. The strength of the SVM classifier has been demonstrated in many research areas such as handwriting recognition ap-

plication, which is described in Dong *et al.* [16,17]. The classifier we use is a modified version of the SVM classifier proposed in [18].

3 Results

To evaluate the proposed method, both chest CT scans (two dimensional and three dimensional images) and dental X-ray images are used to test the proposed method.

3.1 Chest CT Scans

Two Dimensional Scans. Figs. 6 and 7 show the results of two dimensional image segmentation. Fig. 6 shows the results of pathological variational level set segmentation which divides the image into four regions of background, the skeletal structure (bone), the fatty tissue, and the muscle and visceral tissue, as defined in section 2.3. However the variational level set method is a time consuming method which generally takes longer than 10 minutes to segment a 256×256 image for a PC (Pentium 1G Hz and 1GRAM). Moreover, for some cases, level set methods, especially for coupled level set methods, may not converge for some initial curves as pointed out in [3, 19, 20] which limit the usage of the level set method in clinical image processing which has high requirements on speed and robustness. Fig. 7 demonstrates the segmentation results using the proposed method which just takes around 1 second.

Three Dimensional Scans. Figs. 8 and 9 show results on three dimensional image segmentation. Fig. 8 shows variational level set segmentation on volumetric CT scan image ($256 \times 256 \times 100$) which usually takes longer than 2 hours while with our proposed method takes around 20 seconds.

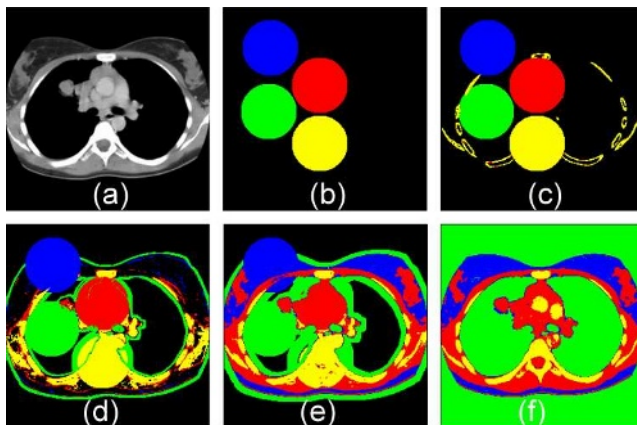


Fig. 6. Experimental Results on CT scans. (a) Iteration 0. (b) Iteration 20. (c) Iteration 50. (d) Iteration 100

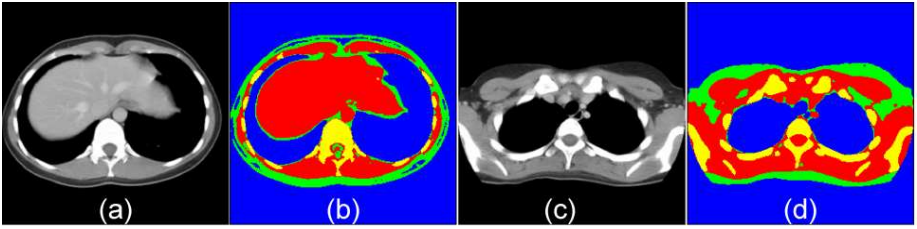


Fig. 7. Experimental Results on CT scans. (a) (c) Original images. (b) (d) Segmented images

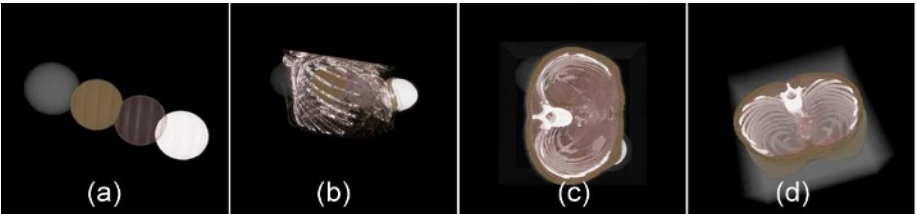


Fig. 8. Volumetric coupled level set segmentation results. (a) Iteration 0. (b) Iteration 30. (c) Iteration 80. (d) Iteration 120

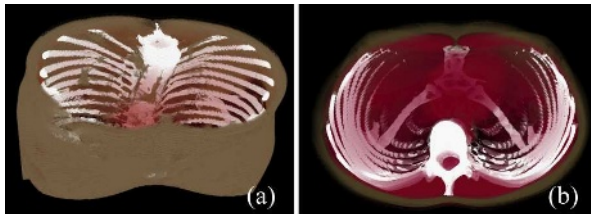


Fig. 9. Volume rendering of segmentation results of using proposed method on chest CT scans. (a) One View. (b) Another view

3.2 Dental X-Ray Images

Dental X-ray segmentation is a challenging problem for classic methods due to the following characteristics: (1) poor image modalities: noise, low contrast, and sample artifacts; (2) complicated topology; and (3) there may not be clear edges between regions of interest which is especially true for dental images with early stage problem teeth. Fig. 10 demonstrates the variational level set segmentation described in section 2.3 on dental X-ray images. As can be seen, the variational level set method is able to successfully segment with the given pathological modelling which provides automatic feature extraction for PCA and SVM training. Fig. 11 shows the results by the proposed method. Since pathological modelling explicitly incorporates regions of problems as part of the modelling, the identification of such areas is an automatic byproduct of the segmentation.

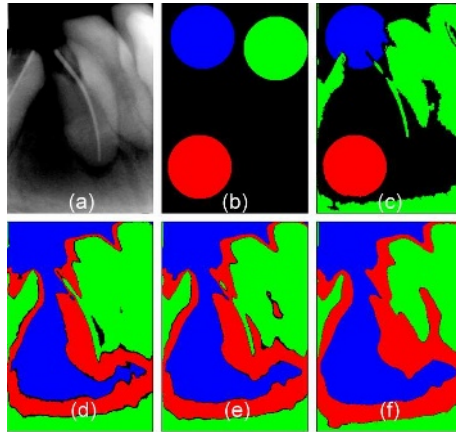


Fig. 10. Coupled level sets segmentation. (a) Iteration 0. (b) Iteration 100. (c) Iteration 500. (d) Iteration 2000. (f) Iteration 2500

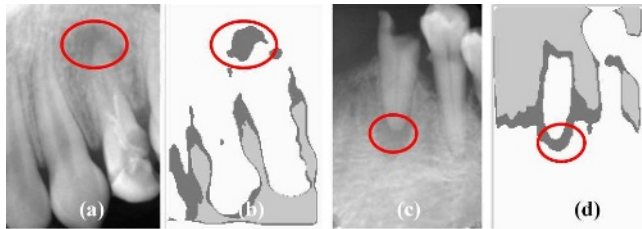


Fig. 11. Experimental Results on Dental X-rays. (a) (c) Original image with problem area circled by dentist. (b) (d) Segmented image

4 Conclusion

This paper proposes a general automatic clinical image segmentation method. The proposed segmentation method contains two stages: a learning stage and a clinical segmentation stage. During the learning stage, manually chosen representative images are segmented using a variational level set method driven by a pathologically modelled energy functional. Then a window-based feature is extracted from the segmented images and the principal component analysis is applied to those extracted features. These results are used to train a support vector machine classifier. During the segmentation stage, the clinical images are classified with the trained SVM. The proposed method takes the strengths of newly developed machine learning and the variational level set methods while limiting their weaknesses to achieve a automatic and fast clinical segmentation. The method is tested with both chest CT scans and dental X-ray images. These results show that the proposed method is able to provide a fast and robust clinical

cal image segmentation of both 2D and 3D images. Due to the use of pathological modelling to define the regions of interest, the segmentation results can be used to further analyze the image. The proposed method can be used as pre-processing step for automatic computer aided diagnosis. We are currently studying other machine learning algorithms for the analysis of segmented images to provide further improved assistance to the doctor or clinician.

References

1. S. Wang, W. Zhu, and Z.-P. Liang, "Shape deformation: SVM regression and application to medical image segmentation," in *ICCV*, pp. 209–216, 2001.
2. S. Li, T. Fevens, and A. Krzyżak, "An SVM based framework for autonomous volumetric medical image segmentation using hierarchical and coupled level sets," in *Computer Aided Radiology and Surgery (CARS)*, (Chicago, USA), pp. 207–212, 2004. 10 Li et al.
3. S. Li, T. Fevens, and A. Krzyżak, "Image segmentation adapted for clinical settings by combining pattern classification and level sets," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, (St-Malo, France), pp. 160–167, 2004.
4. L. M. Lorigo, W. E. L. Grimson, O. D. Faugeras, R. Keriven, R. Kikinis, A. Nabavi, and C.-F. Westin, "Codimension-two geodesic active contours for the segmentation of tubular structures," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2000.
5. S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.
6. R. Kimmel and A. M. Bruckstein, "Regularized Laplacian zero crossings as optimal edge integrators," *International Journal of Computer Vision*, vol. 53, pp. 225–243, July 2003.
7. A. Vasilevskiy and K. Siddiqi, "Flux maximizing geometric ow," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1565–1578, 2002.
8. S. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, energy/bayes/mdl for multiband image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19(9), pp. 884–900, 1996.
9. V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours," *Numer. Math.*, vol. 66, pp. 1–31, 1993.
10. S. Li, T. Fevens, A. Krzyżak, and S. Li, "Level set segmentation for computer aided dental X-ray analysis," in *SPIE Conference on Medical Imaging*, (San Diego, USA), 2005. Accepted.
11. T. Chan and L. Vese, "Active contour model without edges," *IEEE Trans. on Image Processing*, vol. 24, pp. 266–277, 2001.
12. L. Vese and T. Chan, "A multiphase level set framework for image segmentation using the mumford and shah model," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271–293, 2002.
13. C. Samson, L. Blanc-Fraud, G. Aubert, and J. Zerubia, "A level set model for image classification," *International Journal of Computer Vision*, vol. 40, no. 3, pp. 187–197, 2000.
14. M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, (Hawaii), 1991.

15. M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
16. J. Dong, A. Krzyżak, and C. Y. Suen, "A fast parallel optimization for training support vector," in *International Conference on Machine Learning and Data Mining (MLDM)* (P. Perner and A. Rosenfeld, eds.), vol. LNAI 2734, (Leipzig, Germany), pp. 96–105, Springer Lecture Notes in Artificial Intelligence, July 2003.
17. J. Dong, A. Krzyżak, and C. Y. Suen, "Fast SVM training algorithm with decomposition on very large training sets," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2005. to appear.
18. C.-C. Chang and C.-J. Lin, "Training nu-support vector classifiers: theory and algorithms," *Neural Computation*, vol. 13, no. 9, pp. 2119–2147, 2001.
19. M. Holtzman-Gazit, D. Goldsher, and R. Kimmel, "Hierarchical segmentation of thin structure in volumetric medical images," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, (Montreal), 2003.
20. A. Tsai, A. Yezzi, and A. S. Willsky, "Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and magnification," *IEEE Trans. on Image Processing*, vol. 10, pp. 1169–1186, 2001.

Improved MRI Mining by Integrating Support Vector Machine Priors in the Bayesian Restoration

D.A. Karras¹, B.G. Mertzios², D. Graveron-Demilly³, and D. van Ormondt⁴

¹Chalkis Institute of Technology, Dept. Automation and Hellenic Open University,
Rodu 2, Ano Iliupolis, Athens 16342, Greece, Fax: +30 210 9945231
dakarras@teihal.gr, dakarras@ieee.org

²Thessaloniki Institute of Technology, Thessaloniki, Hellas and Democritus University,
Laboratory of Automatic Control Systems, Xanthi, Hellas

³Laboratoire de RMN, CNRS, UPRESA 5012, Universite LYON I-CPE, France

⁴Delft University of Technology, Applied Physics Department,
P.O Box 5046, 2600 GA Delft, The Netherlands

Abstract. The goal of this paper is to present the development of a new image mining methodology for extracting Magnetic Resonance Images (MRI) from reduced scans in k -space. The proposed approach considers the combined use of Support Vector Machine (SVM) models and Bayesian restoration, in the problem of MR image mining from sparsely sampled k -space, following several different sampling schemes, including spiral and radial. Effective solutions to this problem are indispensable especially when dealing with MRI of dynamic phenomena since then, rapid sampling in k -space is required. The goal in such a case is to make measurement time smaller by reducing scanning trajectories as much as possible. In this way, however, underdetermined equations are introduced and poor image extraction follows. It is suggested here that significant improvements could be achieved, concerning quality of the extracted image, by judiciously applying SVM and Bayesian estimation methods to the k -space data. More specifically, it is demonstrated that SVM neural network techniques could construct efficient priors and introduce them in the procedure of Bayesian restoration. These Priors are independent of specific image properties and probability distributions. They are based on training SVM neural filters to estimate the missing samples of complex k -space and thus, to improve k -space information capacity. Such a neural filter based prior is integrated to the maximum likelihood procedure involved in the Bayesian reconstruction. It is found that the proposed methodology leads to enhanced image extraction results favorably compared to the ones obtained by the traditional Bayesian MRI reconstruction approach as well as by the pure Neural Network (NN) filter based reconstruction approach.

Keywords: MRI Reconstruction, MRI Mining, SVM, MLP, Bayesian Restoration.

1 Introduction

A data acquisition process is needed to form the MR images. Such data acquisition occurs in the spatial frequency (k -space) domain, where sampling theory determines

resolution and field of view, and it results in the formation of the k -space matrix. Strategies for reducing image artifacts are often best developed in this domain. After obtaining such a k -space matrix, image reconstruction involves fast multi-dimensional Inverse Fourier transforms, often preceded by data interpolation and re-sampling.

Sampling the k -space matrix occurs along suitable trajectories [1,2,3]. Ideally, these trajectories are chosen to completely cover the k -space according to the Nyquist sampling criterion. The measurement time of a single trajectory can be made short. However, prior to initiating a trajectory, return to thermal equilibrium of the nuclear spins needs to be awaited. The latter is governed by an often slow natural relaxation process that is beyond control of the scanner and impedes fast scanning. Therefore, the only way to shorten scan time in MRI when needed, as for instance in functional MRI, is to reduce the overall waiting time by using fewer trajectories, which in turn should individually cover more of k -space through added curvatures. Although, however, such trajectory omissions achieve the primary goal, i.e. more rapid measurements, they entail undersampling and violations of the Nyquist criterion thus, leading to concomitant problems for image reconstruction.

The above mentioned rapid scanning in MRI problem is highly related with two other ones. The first is the selection of the optimal scanning scheme in k -space, that is the problem of finding the shape of sampling trajectories that more fully cover the k -space using fewer trajectories. Mainly three such alternative shapes have been considered in the literature and are used in actual scanners, namely, Cartesian, radial and spiral [1], associated with different reconstruction techniques. More specifically, the Cartesian scheme uses the inverse 2D FFT, while the radial and spiral scanning involve the Projection Reconstruction, the linogram or the SRS-FT approaches [1,2,3].

The second one is associated with image estimation from fewer samples in k -space, that is the problem of omitting as many trajectories as possible without attaining worse reconstruction results. The main result of such scan trajectories omissions is that we have fewer samples in k -space than needed for estimating all pixel intensities in image space. Therefore, there is infinity of MRI images satisfying the sparse k -space data and thus, the image mining problem becomes ill-posed. Additionally, omissions usually cause violation of the Nyquist sampling condition. Despite the fact that solutions are urgently needed, in functional MRI for instance, very few research efforts exist in the literature. The most obvious and simplest such method is the so called “zero-filling the k -space”, that is, all missing points in k -space acquire complex values equal to zero. Subsequently, image mining is achieved as usually, by applying the inverse Fourier transform to the corresponding k -space matrix. Instead of zero-filling the k -space or using linear estimation techniques [2,3] it might be more advantageous to interpolate it by using nonlinear interpolation procedures, like Artificial Neural Networks (ANN). The Bayesian reconstruction approach, developed by two of the authors [1], briefly presented in the next section is another alternative solution. Such a solution could yield good results concerning MR image mining performance [1]. The main contribution, however, of this paper is to develop a novel MR image mining methodology by involving both Bayesian and Neural restoration (based on SVM approximation) techniques and present its competence and advantages over other rival approaches.

2 The Bayesian MRI Restoration Approach

The Bayesian restoration approach proposed by two of the authors [1], attempts to provide solutions through regularizing the problem by invoking general prior knowledge in the context of Bayesian formalism. The algorithm amounts to minimizing the following objective function [1], by applying the conjugate gradients method,

$$|\underline{\mathbf{S}} - \mathbf{T} \underline{\mathbf{I}}|^2 / (2\sigma^2) + (3/2) \sum_{x,y} \log \{ \alpha^2 + ({}^x\Delta_{xy})^2 + ({}^y\Delta_{xy})^2 \} \quad (1)$$

with regards to $\underline{\mathbf{I}}$, which is the unknown image to be reconstructed that fits to the sparse k-space data given in $\underline{\mathbf{S}}$. The first term comes from the likelihood term and the second one from the prior knowledge term of the Bayesian formulation [1]. The parameter σ amounts to the variance encountered in the likelihood term and in the herein conducted simulations is set equal to 1. In the above formula, $T((k_x, k_y), (x, y)) = e^{-2\pi i(xk_x + yk_y)}$ represents the transformation from image to k-space data (through 2-D FFT). The second term symbols arise from the imposed 2D Lorentzian prior knowledge. ${}^x\Delta_{xy}$ and ${}^y\Delta_{xy}$ are the pixel intensity differences in the x- and y- directions respectively and α is a Lorentz distribution-width parameter. Assuming that $P(\mathbf{I})$ is the prior, imposing prior knowledge conditions for the unknown MRI image, then, the second term of (1) comes as follows.

The starting point is that $P(\mathbf{I})$ could be obviously expanded into $P(\mathbf{I}) = P(I_{0,0}) P(I_{1,0} | I_{0,0}) P(I_{2,0} | I_{0,0}, I_{1,0}) \dots$. If, now, it is assumed that the intensity $I_{x,y}$ depends only on its left neighbor ($I_{x-1,y}$), then the previous $P(\mathbf{I})$ expansion takes on the form $P(\mathbf{I}) = \prod_{(x,y)} P(I_{x,y} | I_{x-1,y})$, provided that the boundaries are ignored. Next, we assume that $P(I_{x,y} | I_{x-1,y})$ is a function only of the difference between the corresponding pixels. This difference is written down as ${}^x\Delta_{xy} = I_{x,y} - I_{x-1,y}$. It has been shown that the probability density function of ${}^x\Delta_{xy}$ is Lorentzian shaped (see [1,2,3]). These assumptions and calculations lead to computing the prior knowledge in the Bayesian reconstruction as in the second term of (1).

Although this Bayesian restoration approach tackles the problem of handling missing samples in k-space, it exhibits, however, the disadvantage that assumes the existence of special probability distributions, given in closed form descriptions, for representing the unknown ones occurred in MRI, which is an issue under question. In this paper we attempt to remedy this problem by proposing additional priors in the Bayesian formulation in order to capture the probability distribution functions encountered in MRI. These priors are constructed through applying a specifically designed Support Vector Machine (SVM) neural filter for interpolating the sparsely sampled k-space.

3 Design of SVM Neural Network Priors

The method herein suggested for designing efficient Priors for the Bayesian reconstruction formalism, is based on the attempt to extract prior knowledge from the process of filling in the missing complex values in k-space from their neighboring complex values. Thus, instead of assuming a Lorentzian prior knowledge to be extracted from the neighboring pixel intensities in MRI, as a constraint to be applied

in the conjugate gradient based Bayesian reconstruction process, the proposed strategy doesn't make any assumption. Instead, it aims at extracting priors without any specific consideration concerning the shape of the distributions involved, by transforming the original reconstruction problem into an approximation one in the complex domain. While linear interpolators have already been used in the literature [2,3], ANN models could offer several advantages when applied as sparsely sampled k-space interpolators. The methodology to extract prior knowledge by applying the ANN filters in MRI reconstruction is described in the following paragraphs.

Step 1. We compile a set of R representative $N \times N$ MRI images with k-space matrices completely known, which comprise the training set of the SVM approximators. Subsequently, we scan these matrices following the specific sampling schemes mentioned above and then, by randomly omitting trajectories the sparse k-spaces are produced, in order to simulate the real MR data acquisition process.

Step 2. The original k-space matrix as well as its corresponding sparse k-space matrix associated with one $N \times N$ MRI training image, is raster scanned by a $(2M+1) \times (2M+1)$ sliding window containing the associated complex k-space values. The estimation of the complex number in the center of this window from the rest of the complex numbers comprising it is the goal of the proposed approximation procedure. Each position of this sliding window is, therefore, associated with a desired output pattern comprised of the complex number in the original k-space corresponding to the window position, and an input pattern comprised of the complex numbers in k-space corresponding to the rest $(2M+1) \times (2M+1) - 1$ window points.

Step 3. Each such pattern is then, normalized according to the following procedure. First, the absolute values of the complex numbers in the input pattern are calculated and then, their average absolute value $|z_{aver}|$ is used to normalize all the complex numbers belonging both in the input and the desired output patterns. That is, if z_1 is such a number then this normalization procedure transforms it into the $z_1/|z_{aver}|$. In the case of test patterns we apply the same procedure. That is, the average absolute value $|z_{aver}|$ for the complex numbers z_i of the test input pattern is first calculated. Then, the normalized complex values $z_i/|z_{aver}|$ feed the SVM approximation filter to predict the sliding window central normalized complex number z_{centre}^{norm} . The corresponding unnormalized complex number is simply $z_{centre}^{norm} * |z_{aver}|$.

Step 4. The next step is the production of training patterns for the SVM approximators and their training procedure. To this end, by randomly selecting sliding windows from the associated k-spaces of the R training images and producing the corresponding input and desired output training pairs of patterns, as previously defined, we construct the set of training patterns. The assumption underlying such an approach of training SVM approximators is that there are regularities in every k-space sliding window, the same for any MRI image, to be captured by the SVMs without any prior assumption for the probability distributions. SVM training is defined by applying the following procedure.

SVMs is a Neural Network (NN) methodology, introduced by Vapnik in 1992 [5]. They have recently started to be involved in many different classification tasks with success. Few research efforts, however, have used them in nonlinear regression tasks as the MR image mining problem we herein present. One of the goals of the herein

study was to evaluate the SVM for Nonlinear Regression approach in such tasks in comparison with other ANN techniques and linear estimation methodologies. The results herein obtained justify that the SVM approach could widely and successfully be involved in function approximation/regression tasks. The task of nonlinear regression using a Support Vector Machine could be defined as follows.

Let $f(\mathbf{X})$ be a multidimensional scalar valued function to be approximated, like the real/imaginary part of the sliding window central normalized complex number $z^{\text{norm}}_{\text{centre}}$ above defined in step 3. Then, a suitable regression model to be considered is:

$$D = f(\mathbf{X}) + n$$

where, \mathbf{X} is the input vector comprised of $(2M+1) \times (2M+1) - 1$ real/imaginary parts of the complex k-space normalized values associated with the window whose central normalized complex value is $z^{\text{norm}}_{\text{centre}}$, n is a random variable representing the noise and D denoting a random variable representing the outcome of the regression process. Given, also, the training sample set $\{(\mathbf{X}_i, D_i)\} (i=1, \dots, N)$ then, the SVM training can be formulated as the optimization problem next outlined:

Find the Lagrange Multipliers $\{\lambda_i\} (i=1, \dots, N)$ and $\{\lambda'_i\} (i=1, \dots, N)$ that maximize the objective function,

$$Q(\lambda_i, \lambda'_i) = \sum_{i=1..N} D_i (\lambda_i - \lambda'_i) - e \sum_{i=1..N} (\lambda_i + \lambda'_i) - 1/2 \sum_{i=1..N} \sum_{j=1..N} (\lambda_i - \lambda'_i) (\lambda_j - \lambda'_j) K(\mathbf{X}_i, \mathbf{X}_j)$$

subject to the constraints:

$\sum_{i=1..N} (\lambda_i - \lambda'_i) = 0$ and $0 \leq \lambda_i \leq C$, $0 \leq \lambda'_i \leq C$ for $i=1..N$, where C a user defined constant.

In the above definition, $K(\mathbf{X}_i, \mathbf{X}_j)$ are the kernel functions. In the problem at hand we have employed the radial basis kernel

$$K(\mathbf{X}, \mathbf{X}_j) = \exp(-1/2\sigma^2 \|\mathbf{X} - \mathbf{X}_j\|^2)$$

Taking into account the previous definitions we can then, fully determine the approximating function as

$$F(\mathbf{X}) = \sum_{i=1..N} (\lambda_i - \lambda'_i) K(\mathbf{X}, \mathbf{X}_i)$$

which estimates the real and the imaginary part of the complex number $z^{\text{norm}}_{\text{centre}}$.

Namely, $F_{\text{real}}(\mathbf{X}_{\text{real}}) = \sum_{i=1..N} (\lambda_{i_{\text{real}}} - \lambda'_{i_{\text{real}}}) K_{\text{real}}(\mathbf{X}_{\text{real}}, \mathbf{X}_{i_{\text{real}}})$ and $F_{\text{imaginary}}(\mathbf{X}_{\text{imaginary}}) = \sum_{i=1..N} (\lambda_{i_{\text{imaginary}}} - \lambda'_{i_{\text{imaginary}}}) K_{\text{imaginary}}(\mathbf{X}_{\text{imaginary}}, \mathbf{X}_{i_{\text{imaginary}}})$ are the two corresponding SVMs. The former is applied to approximate the real part of $z^{\text{norm}}_{\text{centre}}$ while the latter for approximating its imaginary part.

Step 5. After training phase completion, the SVM filter has been designed and can be applied to any similar test MRI image as follows. To this end, the $(2M+1) \times (2M+1)$ sliding window raster scans the sparse k-space matrix associated with this test image, starting from the center. Its central point position moves along the perimeter of rectangles covering completely the k-space, having as center of gravity the center of the k-space array and having distance from their two adjacent ones of 1 pixel. It can move clockwise or counterclockwise or in both directions. For every position of the sliding window, the corresponding input pattern of $(2M+1) \times (2M+1) - 1$ complex

numbers is derived following the above described normalization procedure. Subsequently, this normalized pattern feeds the SVM approximator. The wanted complex number corresponding to the sliding window center, is found as $z_{\text{centre}} = z_{\text{SVM}}^{\text{out}} * |z_{\text{aver}}|$, where $z_{\text{SVM}}^{\text{out}}$ is the SVM output and $|z_{\text{aver}}|$ the average absolute value of the complex numbers comprising the un-normalized input pattern. For each rectangle covering the k-space, the previously defined filling in process takes place so that it completely covers its perimeter, only once, in both clockwise and counterclockwise directions. The final missing complex values are estimated as the average of their clockwise and counter-clockwise obtained counterparts. The outcome of the SVM filter application is the reconstructed test image, herein named SVM_Img (equation (2) below). Its difference from the image $I^{(t)}$ obtained during the previous step of conjugate gradient optimization in the Bayesian reconstruction formula (1), provides the neural prior to be added for the current optimization step.

4 Incorporation of SVM Neural Prior Knowledge into the Bayesian Formalism

Following the 5 steps above, we can formulate the incorporation of SVM priors to the Bayesian restoration process as follows.

- Design the SVM Neural filter as previously defined
- Consider the Bayesian reconstruction formula (1). The image to be optimized is I given the k-space S . The initial image in the process of conjugate gradient optimization is the zero-filled image. At each step t of the process a different $I^{(t)}$ (the image at the t step, that is, the design variables of the problem) is the result. Based on figure 1 below, by applying the SVM filter on the original sparse k-space, but with the missing points initially filled by the FFT of $I^{(t)}$ (in order to derive the $I^{(t)}$ k-space)- and afterwards refined by the SVM predictions, we could obtain the difference $I^{(t)} - \text{SVM_Img}^{(t)}$ as the Neural Prior.

- Therefore, the Neural Network (NN) Prior form, based on SVM approximation is:

$$\sum_{x,y} | \text{SVM_Im } g^{(t)}(x, y) - I^{(t)}(x, y) | \tag{2}$$

where, $\text{SVM_Img}^{(t)}(x,y)$ is the SVM estimated pixel intensity in image space (SVM reconstructed image: Inverse FFT of SVM completed k-space) at step t and $I^{(t)}(x,y)$ is the image obtained at step t of the conjugate gradient optimization process in the Bayesian reconstruction.

- The proposed Prior in the Bayesian reconstruction is given as

$$\text{Final Prior} = \text{Lorentzian Bayesian Prior} + a * \text{SVM_Prior} \tag{3}$$

- That is, the optimization process $I^{(t)}$ is attempted to be guided by the $\text{SVM_Img}^{(t)}$ produced by the SVM. Therefore, the proposed algorithm amounts to minimizing the following objective function, by applying again the conjugate gradients method,

$$\begin{aligned}
 & | \underline{S} - T \underline{I} |^2 / (2\sigma^2) + (3/2) \sum_{x,y} \log \{ \alpha^2 + ({}^x\Delta_{xy})^2 + ({}^y\Delta_{xy})^2 \} + \\
 & a^* \sum_{x,y} | \text{SVM_Im } g^{(t)}(x, y) - I^{(t)}(x, y) |
 \end{aligned} \tag{4}$$

where, a=3/2 is the value used for the parameter a above in our experiments.

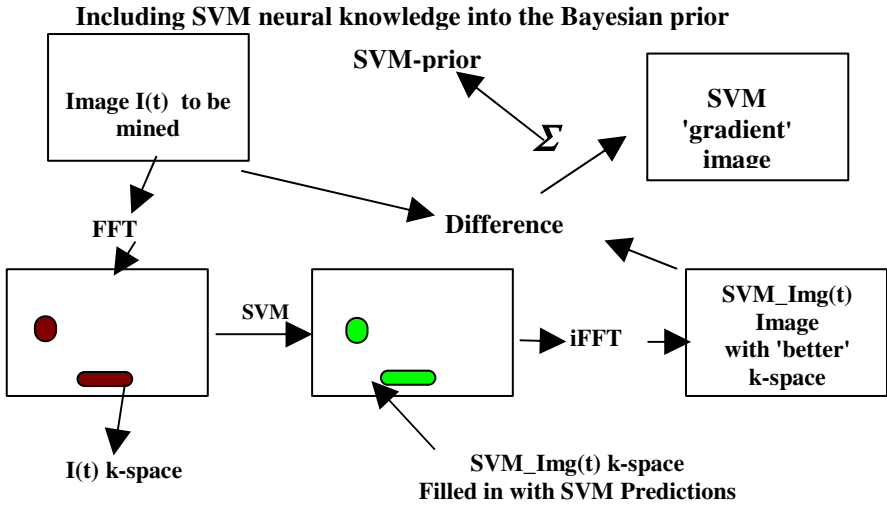


Fig. 1. The difference between the image to be mined I(t) and the SVM mined image SVM_Img(t) constitutes the neural prior

5 Evaluation Study and Conclusions

An extensive experimental study has been conducted in order to evaluate the above defined novel Bayesian reconstruction methodology. All the methods involved have been applied to a large MRI image database downloaded from the Internet, namely, the Whole Brain Atlas <http://www.med.harvard.edu/AANLIB/home.html> (copyright © 1995-1999 Keith A. Johnson and J. Alex Becker). We have used 10 images randomly selected out of this collection for training the SVM approximation filters, and 10 images, again randomly selected for testing the proposed and the rival reconstruction methodologies. All images are 256 by 256. Their k-space matrices have been produced applying the 2D FFT to them. Radial trajectories have been used to scan the resulted 256 X 256 complex k-space arrays. 4 X 256 = 1024 radial trajectories are needed to completely cover such k-spaces. In order to apply the reconstruction techniques involved in this study, each k-space has been sparsely sampled using 128 only radial trajectories. Regarding the sliding window raster scanning the k-space, a 5 X 5 window was the best selection.

Concerning the SVM filter architecture, the 48-17-2 (number of inputs-number of support vectors-number of outputs) one was found after the SVM design stage (step 4,

section 3). Actually, as explained in step 4 of section 3 this SVM approximation filter is comprised of two different SVMs (this explains the number two of outputs). The first one is associated with approximating the real part of $z^{\text{norm}}_{\text{centre}}$ while the second one with approximating its imaginary part. This SVM approximation filter has been trained using 3600 training patterns. The compared reconstruction techniques involved in this study are: the proposed novel Bayesian mining approach, the traditional Bayesian reconstruction technique as well as the SVM filtering approximation technique. In addition, a Multilayer Perceptron (MLP) neural interpolator of 48-12-2 (number of inputs-hidden nodes-number of outputs) architecture (found to be the best one) has been involved in the comparisons. Moreover, the simplest “interpolation” approach, namely filling in the missing samples in k-space with zeroes and then, reconstructing the image, has been invoked. All these methods have been implemented using the MATLAB programming platform.

Concerning the measures involved to quantitatively compare reconstruction performance, we have employed the usually used Sum of Squared Errors (SSE) between the original MRI image pixel intensities and the corresponding pixel intensities of the reconstructed image. Additionally, another quantitative measure has been used, which expresses performance differences in terms of the RMS error in dB [4]:

```
lambda=(image_recon(:)*image_orig(:))/(image_recon(:)*image_recon(:));residu=i
mage_orig-lambda*image_recon;
dB=10*log10((image_orig(:)*image_orig(:))/(residu(:)'*
residu(:)));
```

The quantitative results obtained by the different reconstruction methods involved are outlined in table 1 (average SSE and RMS errors for the 10 test MRI images). Concerning reconstruction performance qualitative results, a sample is shown in figure 2. Both quantitative and qualitative results clearly demonstrate the superiority of the proposed Bayesian image mining methodology embedding SVM filtering based prior knowledge, in terms of MRI image restoration performance over the other rival methodologies (simple Bayesian restoration, SVM / MLP MRI mining filter and zero-filled reconstructions). Future trends of our research efforts include implementation of the 3-D Bayesian reconstruction with Neural Network priors for f-MRI as well as applications in MRI image segmentation for tumor detection.

Table 1. The quantitative results with regards to reconstruction performance of the various methodologies

MRI Mining Method	SSE (average in the 10 test MRI images)	dB (average in the 10 test MRI images)
Proposed Bayesian MR Image mining with SVM Prior	2.63 E3	17.52
Proposed Bayesian MR Image mining with MLP Prior	2.85 E3	16.67
Traditional Bayesian restoration	3.40 E3	15.92
SVM restoration	3.27 E3	16.02
MLP restoration	3.30 E3	15.98
Zero-filling restoration	3.71 E3	15.26

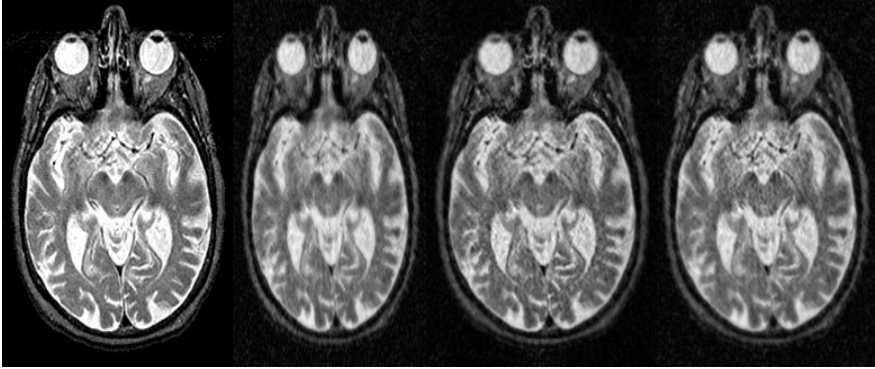


Fig. 2. From left to right: The proposed Bayesian MR Image mining involving SVM priors, the sparsely sampled k-space ($nr=128$)-zerofilled image reconstruction, the MLP filtering and the traditional Bayesian reconstruction results. The Test Image illustrates a brain slice with Alzheimer's disease (<http://www.med.harvard.edu/AANLIB/cases/case3/mr1-tc1/020.html>)

References

- [1] G.H.L.A. Stijnman, D. Graveron-Demilly, F.T.A.W. Wajer and D. van Ormondt: MR Image Estimation from Sparsely Sampled Radial Scans "Proc. ProRISC/IEEE Benelux workshop on Circuits, Systems and Signal Processing, Mierlo (The Netherlands), 1997" , 603-611
- [2] M.R. Smith, et al.: Application of Autoregressive Modelling in Magnetic Resonance Imaging to Remove Noise and Truncation Artifacts, *Magn. Reson. Imaging*, 4, 257 (1986).
- [3] P. Barone and G. Sebastiani: A New Method of Magnetic Resonance Image Reconstruction with Short Acquisition Time and Truncation Artifact Reduction, *IEEE Trans. Med. Imaging*, 11, 250 (1992).
- [4] I. Dologlou, et al.: Spiral MRI Scan-Time Reduction through Non-Uniform Angular Distribution of Interleaves and Multichannel SVD Interpolation, in "Proc. ISMRM, 4th Meeting, N.Y, 1996", p. 1489.
- [5] Haykin S., "Neural Networks: A comprehensive foundation", Second Edition, Prentice Hall, 1999.

Prediction of Secondary Protein Structure Content from Primary Sequence Alone – A Feature Selection Based Approach

Lukasz Kurgan and Leila Homaeian

University of Alberta, Department of Electrical and Computer Engineering,
Edmonton, Alberta, Canada T6G 2V4
{lkurgan, leila}@ece.ualberta.ca

Abstract. Research in protein structure and function is one of the most important subjects in modern bioinformatics and computational biology. It often uses advanced data mining and machine learning methodologies to perform prediction or pattern recognition tasks. This paper describes a new method for prediction of protein secondary structure content based on feature selection and multiple linear regression. The method develops a novel representation of primary protein sequences based on a large set of 495 features. The feature selection task performed using very large set of nearly 6,000 proteins, and tests performed on standard non-homologues protein sets confirm high quality of the developed solution. The application of feature selection and the novel representation resulted in 14-15% error rate reduction when compared to results achieved when standard representation is used. The prediction tests also show that a small set of 5-25 features is sufficient to achieve accurate prediction for both helix and strand content for non-homologous proteins.

1 Introduction

In the recent years increasing knowledge of protein structure accelerated medical research. Research in protein structure and interactions is of paramount importance to modern medicine, as it enhances general understanding of biological processes, and protein functions in particular. One of the most important related applied research and development areas is rational drug design, which aims to cut down costs and accelerate development process of drugs based on analytical models.

Protein structure can be learned by experimental and computational procedures. This paper develops a new computational method for prediction of protein secondary content. It proposes to perform prediction based on combination of feature selection procedure and data mining based approach. The proposed method extends the existing prediction methods by using a novel representation of primary protein structure. Comprehensive feature selection procedure performed with a very large set of almost 6,000 proteins resulted in development of an accurate prediction method that reduced error rates by 14-15% when compared to commonly used feature representation. Independent prediction on non-homogenous protein sets show that a small set of 5-25 features is sufficient to achieve high quality prediction models.

In general, protein structure can be described on three levels: primary structure (Amino Acid (AA) sequence also called primer), secondary structure (folding of the primer into two-dimensional shapes, such as helices, strands, and various coils or turns), and tertiary structure (folding of the two-dimensional shapes into three-dimensional molecule). The Dictionary of Secondary Structures of Proteins annotates each AA as belonging to one of eight secondary structure types [4], which are typically reduced to three groups: helix, strand, and coil. The primary structure is currently publicly known for hundreds of thousands of proteins, e.g. NCBI protein database contains approximately 2 millions proteins, and SWISS-PROT database [3], stores over 159K primers. The secondary and tertiary structure is known for relatively small number of proteins, i.e. the Protein Data Bank (PDB) [1], currently contains about 30K proteins, out of which only a small portion have correct secondary structure and tertiary structure information. At the same time research in protein interactions and functions requires knowledge of tertiary structure. Experimental methods for discovery of secondary and tertiary structure such as X-ray crystallography and nuclear magnetic resonance spectroscopy are time consuming, labor expensive, and cannot be applied to some proteins [6]. Computational methods perform prediction of the tertiary structure with an intermediate step of predicting the secondary structure.

Computational methods for prediction of secondary structure from the primary sequence aim to close the existing gap between the number of known primary sequences and higher structures. One of the important pieces of information to perform prediction of secondary structure is protein content. While the secondary structure prediction aims to predict one of the three groups for each AA in the primary sequence, the secondary content prediction methods aim to predict amount of helix and strand structures in the protein. The secondary structure content can be learned experimentally by using spectroscopic methods, such as circular dichroism spectroscopy in the UV absorption range [13], and IR Raman spectroscopy [2]. Unsatisfactory accuracy and inconvenience of the experimental methods in some cases makes the computational approaches worth pursuing [20]. Computational methods have long history, and usually used statistical methods and information about AA composition of proteins to perform prediction.

This paper describes a novel approach that considers two aspects of content prediction task: quality of primary sequence representation and design of a prediction method. The existing methods, one the other hand, applied different prediction methods, but concentrated only on one dominant AA sequence representation. Secondary content prediction consists of two steps. First, primary sequence is converted into feature space representation, and next the helix and strand content are predicted using the feature values. A typical feature space representation consists of composition vector, molecular weight, and structural class, which are explained later. The first content prediction effort was undertaken in 1973 and used Multiple Linear Regression (MLR) method to predict content based on the composition vector [8]. A number of approaches, which used some combination of the composition vector, molecular weight, and structural class representation and neural network [10], analytic vector decomposition technique [5], and MLR method [17] [18] [19] [20] to predict the content were developed. A novel method that uses both composition vector and composition moment vector and a neural network was recently developed [12].

2 Proposed Prediction Method

The main difference between proposed and existing methods lies in the feature space representation used for prediction. The new method considers a large and diverse set of features, and performs feature selection to find optimal, in terms of quality of prediction and number of used features, representation. The existing methods consider very limited feature representation. After optimal representation is selected, the new method uses the most popular MLR for prediction of the content, see Figure 1.

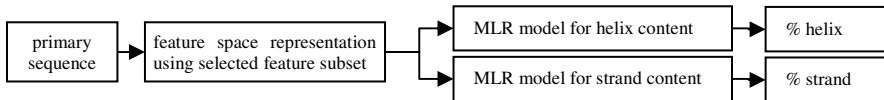


Fig. 1. Procedure for prediction of helix and strand content

The prediction is usually performed with an intermediate step when primary sequence is converted into feature space representation. The existing content prediction methods use a limited set of features while other methods, such as for prediction of protein structure or function, use a more diverse and larger number of features. This paper investigates if a more diverse set of features would help in content prediction. The considered set of features is summarized in Table 1 and later explained in detail.

Table 1. Features used to describe primary protein sequence and their applications

Feature	application type	reference(s)
Protein sequence length, avg molecular weight, avg isoelectric point	protein content and function prediction	[10] [14]
Composition vector	protein structure and content prediction	[5] [8] [10] [12] [17] [18] [19] [20]
1 st order composition moment vector	protein content prediction	[12]
2 nd order composition moment vector		
R-groups	protein structure and content prediction	[11]
Exchange groups	protein family and structure prediction	[15] [16]
Hydrophobicity groups	protein function prediction, structural and functional relationships	[7] [9] [14]
Electronic groups	protein structure prediction	[6]
Chemical groups	protein structure prediction	[6]
Other groups	protein function prediction, structural and functional relationships	[7] [14]
Dipeptides	protein function prediction	[14]

The properties include length, weight, and average isoelectric point. Protein length is defined as the number of AAs. To compute the molecular weight, the residue average weight values are summed and a water molecule mass is added. Average isoelectric point is computed using average isoelectric point values of all AAs in the primer; values are available at www.ionsource.com/virtit/VirtualIT/aainfo.htm. These features were used for protein content and function prediction [10] [14]. Composition vector is defined as composition percentage of each AA in the primary sequence. Composition moment vector takes into account position of each AA in the primary sequence [12]:

$$x_i^{(k)} = \frac{\sum_{j=1}^{K_i} n_{ij}^k}{N(N-1)\dots(N-k)} \tag{1}$$

where n_{ij} and x_i represent the j^{th} position of the i^{th} AA, and the composition of the i^{th} AA in the sequence, respectively; k is the order of the composition moment vector.

The 1st and 2nd orders were used, while the 0th order reduces to the composition vector. Composition vector was used extensively for both protein structure and content prediction [5] [8] [10] [12] [17] [18] [19] [20], while composition moment vector was recently proposed for protein content prediction [12]. The property groups divide the AA into groups related to specific properties of individual AAs or entire protein molecule. Several different properties, such as hydrophobicity, pI, electric charge, chemical composition, etc., that are summarized in Tables 2 and 3 are considered.

Table 2. Property based AA groups

Groups	Subgroups	AAs	Groups	Subgroups	AAs
R-group	Nonpolar aliphatic	AVLIMG	Hydrophobicity group	Hydrophobic	VLIMAFPWYCG
	Polar uncharged	SPTCNQ		Hydrophilic basic	KHR
	Positively charged	KHR		Hydrophilic acidic	DE
	Negative	DE		Hydrophilic polar with uncharged side chain	STNQ
	Aromatic	FYW			
Exchange group	(A)	C	Electronic group	Electron donor	DEPA
	(C)	AGPST		Weak electron donor	VLI
	(D)	DENQ		Electron acceptor	KNR
	(E)	KHR		Weak electron acceptor	FYMTQ
	(F)	ILMV		Neutral	GHWS
	(G)	FYW		Special AA	C
Other group	Charged	DEKHRVLI	Other group	Tiny	AG
	Polar	DEKHRNTQSYW		Bulky	FHWYR
	Aromatic	FHWY		Polar uncharged	NQ
	Small	AGST			

R-group combine hydrophathy index, molecular weight and pI value together [11]. Exchange group represent conservative replacements through evolution. Hydrophobicity groups divide AAs into hydrophobic, which are insoluble or slightly soluble in water, in contrast with hydrophilic, which are water-soluble. Electronic group divides AAs based on their electronic properties, i.e. if they are neutral, electron donor or electron acceptor. Chemical group is associated with individual AAs. There are 19 chemical groups of which AAs are composed. Some of them are listed in Table 3. Other group considers the following mixed classes: charged, polar, aromatic, small, tiny, bulky, and polar uncharged. For each of the groups, the composition percentage of each subgroup in a protein sequence is computed. We note that these groups were extensively used for protein family, structure, function, prediction and to discover structural and functional relationships between proteins [6] [7] [14] [15] [16]. Finally, dipeptides are simply pairs of adjacent AAs in the primary sequence. The composition

Table 3. Chemical groups for AAs

AA	associated chemical groups
A	CH CO NH CH ₃
C	CH CO NH CH ₂ SH
D	CH CO NH CH ₂ CO COO ⁻
E	CH CO NH CH ₂ CH ₂ CO COO ⁻

percentage of each pair is computed. They were previously used for protein function prediction[14].

2.1 Feature Selection for Protein Secondary Content Prediction

The above features were considered for prediction of protein secondary content. Initially correlation between features was investigated to find out if they are independent. The correlated features must be removed since they cannot be used with MLR model. Several correlated features were discovered. For example, some chemical subgroups were correlated with other features, such as composition vector, R-group, and other subgroups in the chemical group. The reason is that some chemical groups appear only in one AA or a group of AAs for which the composition percentage is computed in another feature. For example, COO⁻ is found only in AAs D and E, which is identical to negative R-group, while some chemical groups always appear in the same AAs, such as C and NH₂. Table 4 shows final set of 495 features after removing overlapping and correlated features and provides abbreviation and indices that are used in the paper.

Table 4. List of features considered for feature selection

Feature	Abbr.	Indices
Protein sequence length	SL	1
Average molecular weight	MW	2
Average isoelectric point	IP	3
Composition vector (in alphabetical order)	CV	4-23
1 st order composition moment vector (alphabetically)	MV1	24-43
2 nd order Composition moment vector (alphabetically)	MV2	44-63
R-groups (<i>AVLIMG, SPTCNQ, KHR, DE, FYW</i>)	RG	64-68
Exchange groups (<i>AGPST, DENQ, ILM</i>)	XG	69-71
Hydrophobicity groups (<i>VLIMAFPWYCG, STNQ</i>)	HG	72-73
Electronic groups (<i>DEPA, LIV, KNR, FYMTQ, GHWS</i>)	EG	74-78
Chemical groups (<i>C, CAROM, CH, CH₂, CH₂RING, CH₃, CHAROM, CO, NH, OH</i>)	CG	79-88
Other groups (<i>DEKHRVLI, DEKHRNTQSYW, FHWY, AGST, AG, FHWYR NQ</i>)	OG	89-95
Dipeptides (alphabetically)	DP	96-495

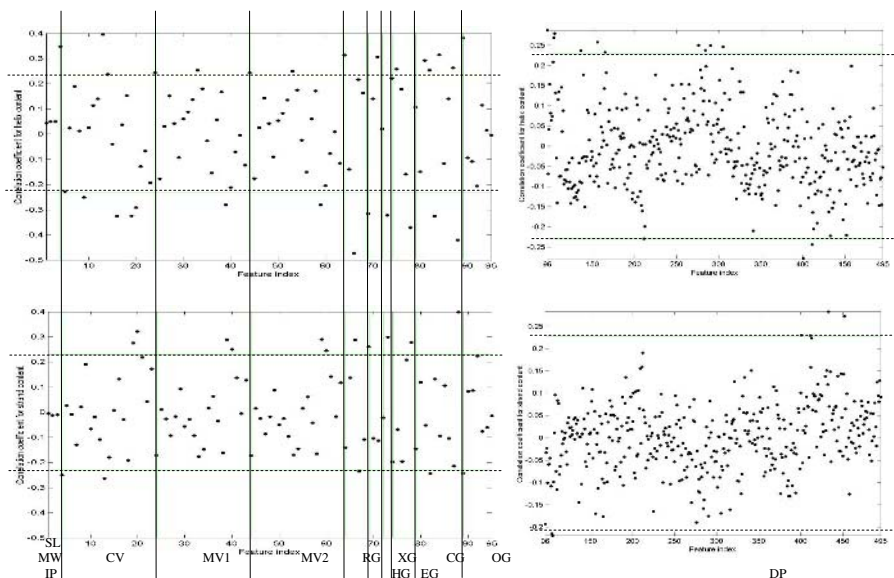


Fig. 2. Results of correlation test for the set of 495 features

Naïve Correlation Based Feature Selection. The simplest feature selection involves computing correlation between a given and the predicted feature and selection of a given number of features with highest correlations. Correlation values of 495 features and the helix and strand content were computed and are summarized in Figure 2.

Analysis of the figure shows that there are no strong correlations that can be used to select a suitable subset of features for content prediction. The strongest correlation values were in 0.3-0.4 range. None of the feature sets, i.e. physical properties, composition and composition moment vectors, property groups and dipeptides, can be evaluated as better or best correlated. The strongest correlated features, using correlation thresholds of 0.229 and -0.229, shown in Figure 2, are given in Table 5.

Table 5. The best correlated feature

structure	correlation	feature /values														
helix	negative	RG ₂	CG ₁₀	EG ₅	CV _P	CG ₅	CV _S	HG ₂	XG ₁	CV _T	MV _{2S}	MV _{1S}	DP _{SG}	CV _G	DP _{SS}	DP _{GS}
		0.47	0.42	0.37	0.32	0.32	0.32	0.32	0.31	0.29	0.28	0.27	0.27	0.25	0.24	0.22
	positive	CV _L	OG ₁	CV _A	CG ₆	RG ₁	XG ₃	CG ₃	DP _{LA}	DP _{AL}	DP _{AK}	CG ₉	EG ₂	DP _{EA}	MV _{1L}	CG ₄
		0.39	0.38	0.34	0.31	0.31	0.30	0.29	0.28	0.27	0.26	0.26	0.25	0.25	0.25	0.25
		DP _{LA}	MV _{2L}	DP _{LR}	DP _{ML}	MV _{2A}	MV _{1A}	DP _{LK}	CV _M	DP _{DA}	DP _{EL}					
		0.25	0.24	0.24	0.24	0.24	0.24	0.23	0.23	0.23	0.23					
strand	negative	CV _L	CV _A	CG ₄	OG ₁	RG ₃										
		0.26	0.25	0.24	0.24	0.23										
	positive	CG ₁₀	CV _T	HG ₂	MV _{2S}	MV _{1S}	RG ₂	DP _{TY}	EG ₅	CV _S	DP _{VT}	XG ₁	MV _{1T}	MV _{2T}	DP _{SG}	
		0.39	0.32	0.29	0.28	0.28	0.28	0.28	0.27	0.27	0.27	0.25	0.24	0.24	0.23	

Correlation accommodates only for correlation between individual features and the predicted values, while more complex correlation that include multiple features together exists. Therefore regression based correlation feature selection was performed.

Regression-Correlation Based Feature Selection. The feature selection was performed according to the procedure shown in Figure 3.

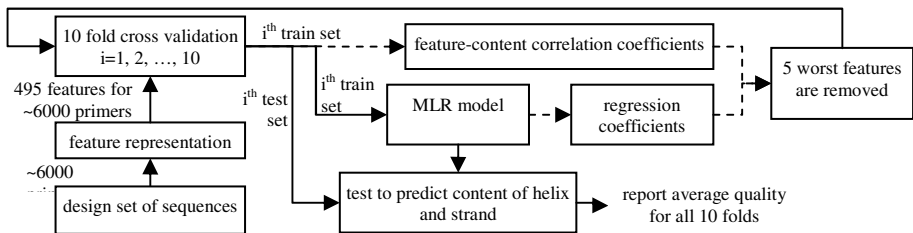


Fig. 3. Feature selection procedure performed independently for helix and strand content

Feature selection is performed independently for helix and strand content prediction. It uses dataset of about 6000 proteins extracted from PDB (described later) to investigate two selection procedures. Each of the 6000 primers is first converted into 495 features representation. Next, the dataset is split in the 10-fold cross validation (10CV) manner, and MLR model is computed for each fold. The model is tested with the test set and average quality over 10 folds is reported. Next, five worst features are

selected and removed, and remaining 490 features are used again to perform MLR. The process repeats until 5 features are left. In each iteration the worst 5 features are selected according to two independent criterions: 1) smallest values of corresponding regression coefficients, 2) smallest values of correlation coefficients between a given feature and the helix/strand content. Both sets of coefficients are recomputed and averaged for each cross-validation fold. The regression coefficients are constants of the MLR model. We assume that the lower the coefficient value the lesser the corresponding feature's impact on the predicted helix or strand content, and therefore the less useful it is for prediction. Similarly correlation coefficients express correlation between a given feature and the predicted helix or strand values. Again, the lower the correlation values the less useful the feature is. The main difference between the coefficient sets is that the MLR considers all features together while the correlation considers each feature independently. The results are discussed in the next section.

3 Experiments

Experiments apply 3 datasets, one for feature selection and two for validation of the developed method and comparison with other prediction methods. The feature selection dataset was extracted from PDB (release as of August 12th 2004) to cover wide range of known proteins. For proteins that have isotopes, the last one was selected. The proteins were filtered according to a set of rules shown in Table 6 to eliminate errors and inconsistent data. Also, sequences with identical primer and different secondary sequences were eliminated. Lastly, sequences with ambiguous AAs in the primer, i.e. B or Z, were removed resulting in a dataset with 5834 sequences that include homologous sequences. The length of the shortest sequence is 6 and of the longest sequence is 1295. The test datasets include:

Table 6. Filters used to derive feature selection dataset

Type of the Problem	# seq	Type of the Problem	# seq
Sequence length < 4	455	Helix indexed out of sequence	10038
Illegal AA	11540	Strand indexed out of sequence	8023
residue called <i>UNK</i>	25	Coil indexed out of sequence	219
More/less residues than the sequence length	9	Overlap of helix and strand	782
Helix of length < 3	1291	Overlap of helix and coil	1342
Strand of length < 2	19022	No secondary structure	9972
		No primary structure	13

- 210 non-homologous proteins set described in [20]. Although these proteins satisfy criteria defined in Table 6, 11 proteins were excluded from experiments, since they include unknown AA X in their primer in the newest PDB release. Therefore 199 proteins were used. The excluded proteins are: 1MBA_, 1MDC_, 1OPAA, 4SBVA, 1FBAA, 1ETU_, 1GP1A, 3ADK_, 1CSEI, 1ONC_, 1FUS_.
- 262 non-homologous proteins set described in [5]. Among the original set only 52 proteins were found in newest PDB release and satisfied criteria from Table 6.

Feature selection was performed using two approaches to select worst performing features for deletion, one based on correlation and the other based on regression coefficients. The content prediction quality was evaluated using two measures [20]:

$$e = \frac{\sum_{k=1}^N |F_k - D_K|}{N}, \quad \sigma = \sqrt{\frac{\sum_{k=1}^N (e - |F_k - D_K|)^2}{N-1}} \quad (2)$$

where e is an average error, σ is standard deviation, F_K is the predicted helix or strand content, D_K is the known content, and N is number of predicted proteins.

Results are shown in Figure 4. Each experiment involves 10CV. Feature selection results are based on computation of about 4000 MLR models. The optimal, in terms of trade-off between error e and number of features, subsets are shown by dashed lines. For both prediction of strand and helix content 4 subsets were selected: for the lowest error value (L), for the last five features (F), for a feature subset of small size (S), and for the best relative ratio between error and feature subset size (M).

The results for selected 4 datasets for both correlation and regression coefficient based approaches and helix and strand prediction are given in Table 7. It shows that minimum error for helix and strand content prediction is 11.28% and 8.67% respectively, and was achieved for regression based selection for dataset L. The maximum error when using just last 5 features is 15.16% and 11.48% for helix and strand

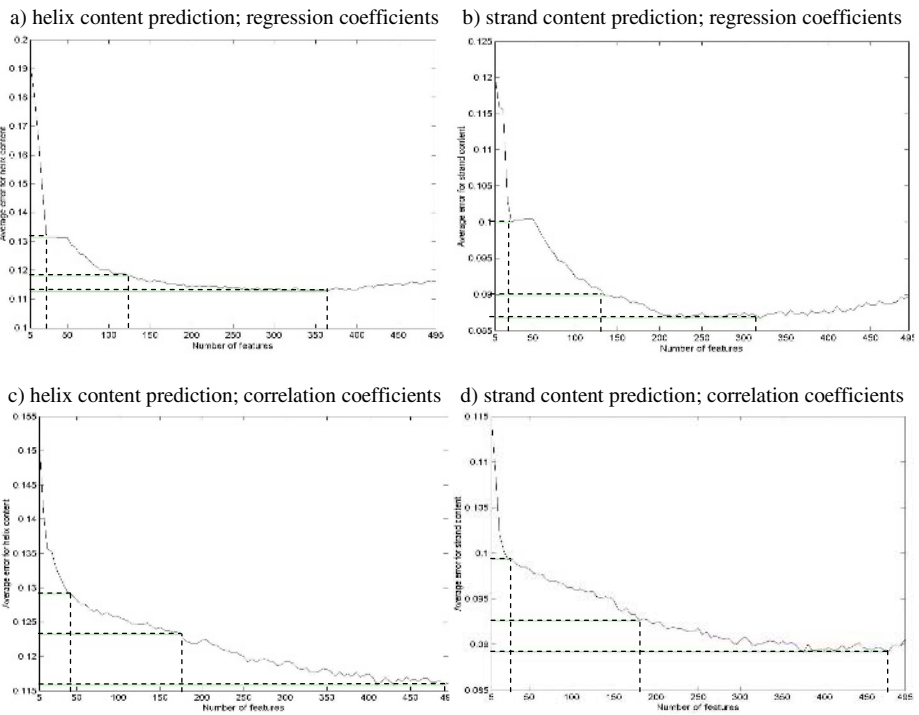


Fig. 4. Results for feature selection experiments using regression and correlation coefficients

prediction respectively, and was achieved for correlation based selection. Therefore 25% error reduction is achieved by using dataset L instead of F. Datasets S and M give relatively good tradeoff between prediction error and number of features. Dataset S with just 25 features for helix prediction gives 12.91% error, while for strand it gives 10% error. Similarly for M dataset, 125 features are used to predict helix content with 11.81% error while 135 features to predict strand content with 8.99% error.

The regression coefficients based selection gives better results for M and L datasets, while correlation coefficients based selection is better for small subsets. This results agrees with our expectations, since regression method benefits from relationships between features, while correlation based method considers each feature independently.

Table 7. Summary of feature selection results (best and worst results are shown in bold; baseline results achieved when composition vector is used are shown in italics)

dataset name			5Features (F)	Small (S)	Medium (M)	Large (L)
regression coeff. selection	helix	# features	5	25	125	365
	prediction	e(σ)	0.1917 (.0171)	0.1315 (.0116)	0.1181 (.0106)	0.1128 (.0102)
	strand	# features	5	25	135	320
	prediction	e(σ)	0.1203 (.0080)	0.1000 (.0067)	0.0899 (.0059)	0.0867 (.0057)
correlation coeff. selection	helix	# features	5	40	175	490
	prediction	e(σ)	0.1516 (.0139)	0.1291 (.0117)	0.1232 (.0111)	0.1158 (.0115)
	strand	# features	5	25	180	475
	prediction	e(σ)	0.1148 (.0074)	0.0994 (.0067)	0.0926 (.0063)	0.0892 (.0063)
composition vector	helix	# features	20	strand	# features	20
	prediction	e(σ)	<i>0.1329</i> (.0117)	prediction	e(σ)	<i>0.1011</i> (.0067)

Another experiment, which involves 10CV prediction using 20 features composition vector to predict the content, was performed, see Table 8. Since composition vector is the most utilized feature set for prediction (all published results use it for prediction [5] [8] [10] [12] [17] [18] [19] [20]) this results gives a baseline to verify that feature selection procedure improves the existing prediction approaches. For the helix prediction a slight improvement of about 0.5% (which translates into 3% error rate reduction) was achieved by using S subsets consisting of 40 features. The 2% error rate improvement (which translates into 15% error rate reduction) was achieved

Table 8. Comparison of error rates for prediction of secondary structure content for different methods and for different considered feature subsets (best results shown in bold; baseline results shown in italics)

method	test dataset (reference)	feature subset	Resubstitution e(σ)		Jackknife e(σ)	
			helix	strand	helix	strand
this paper MLR	199 out of 210 [20]	F _{reg}	0.176 (.017)	0.125 (.007)	0.181 (.018)	0.128 (.008)
		S _{reg}	0.143 (.013)	0.106 (.006)	0.166 (.018)	0.123 (.008)
		M _{reg}	0.092 (.005)	0.052 (.001)	0.263 (.044)	0.178 (.020)
		F _{corr}	0.171 (.016)	0.126 (.007)	0.176 (.017)	0.130 (.007)
		S _{corr}	0.144 (.012)	0.103 (.006)	0.185 (.021)	0.119 (.008)
		M _{corr}	0.051 (.001)	0.030 (.000)	0.514 (.167)	0.354 (.065)
		CV	<i>0.148</i> (.014)	<i>0.110</i> (.005)	<i>0.167</i> (.018)	<i>0.123</i> (.007)
this paper MLR	52 out of 262 [5]	F _{reg}	0.164 (.001)	0.156 (.016)	0.190 (.024)	0.179 (.023)
		S _{reg}	0.115 (.006)	0.098 (.004)	0.240 (.029)	0.208 (.024)
		F _{corr}	0.164 (.014)	0.154 (.012)	0.189 (.021)	0.175 (.017)
		S _{corr}	0.085 (.004)	0.095 (.005)	0.448 (.126)	0.193 (.022)
		CV	<i>0.118</i> (.005)	<i>0.109</i> (.005)	<i>0.211</i> (.022)	<i>0.194</i> (.021)
AVDM-1	262 [5]	CV	0.144 (.117)	0.118 (.096)	0.145 (.017)	0.120 (.097)
AVDM-2		CV	0.132 (.109)	0.114 (.096)	0.142 (.115)	0.124 (.105)
MLR	210 [20]	CV	0.122 (.089)	0.108 (.082)	0.135 (.103)	0.120 (.097)

by using L subset consisting of 265 features. For the strand prediction, the 0.2% error rate improvement was achieved for the 25 features subset, while 1.4% improvement (14% error rate reduction) was achieved when 365 features were used. Although the achieved improvement seem small, the 15% and 14% error rate reduction in medically related field should be perceived as a significant result, especially that it is backed up by a study that considers a large and comprehensive set of proteins.

Prediction tests were performed to test selected feature subsets. Subsets F_{reg} , S_{reg} , and M_{reg} for regression coefficients and F_{corr} , S_{corr} , and M_{corr} for correlation coefficients based selection were used to perform independent test on the test datasets. Prediction of the secondary content was performed using MLR method. In case of regression number of data points (proteins in the dataset) should be larger than number of features. Therefore for 52 protein dataset only F and S subsets were considered.

Test consists of resubstitution and jackknife procedures [20] The first procedure trains and test on the same dataset, while the other is a leave-one-out test. Test results are summarized and compared with other methods in Table 8. The table also includes results for MLR based prediction when the standard composition vector (CV) feature set is used. Since resubstitution test trains and test on the same data, it is prone to overfitting. Thus analysis concentrates on jackknife test results. Baseline results that apply composition vector are always worse than the best results achieved by the generated feature subsets. Subset S_{reg} generates slightly better results for helix prediction, while subset S_{corr} is better in case of strand prediction for the set of 199 proteins. Similarly models generated using subset F_{corr} reduce error rates for both helix and strand prediction by over 10% in case of the 52 protein set (18.9% error rate was achieved for F_{corr} while 21.1% was achieved for composition vector for helix prediction, while 17.5% and 19.4% error rates were achieved for strand prediction respectively). Subsets F that contain only 5 features achieve better results than prediction using 20 features composition vector. The results justify feature selection as a useful method not only to improve prediction results, but also to possibly reduce the number of features necessary for the secondary content prediction. The selected subsets F and S for both correlation and regression based feature selection are listed in Table 9.

Table 9. Selected subsets of features

Data	Struct.	Features
F_{reg}	helix	CV ₁₂ CV ₁₄ CV ₂₀ OG ₃ OG ₇
	strand	CV ₁ CV ₁₁ CV ₁₂ CV ₁₄ OG ₇
S_{reg}	helix	CV ₂ CV ₅ +CV ₁₂ CV ₁₄ CV ₁₇ CV ₁₈ CV ₁₉ CV ₂₀ RG ₁ XG ₂ XG ₃ EG ₁ EG ₂ EG ₃ CG ₆ OG ₂ OG ₃ OG ₆ OG ₇
	strand	CV ₁ +CV ₂₀ RG ₁ RG ₂ RG ₄ RG ₅ XG ₁ XG ₂ XG ₃ HG ₁ HG ₂ EG ₁ +EG ₅ CG ₁ +CG ₁₀ OG ₁ +OG ₇ DP ₁₃ DP ₂₀ DP ₂₂ DP ₂₉ DP ₃₀ DP ₃₁ DP ₃₂ DP ₃₄ DP ₃₅ DP ₃₈ DP ₄₆ DP ₅₈ DP ₆₆ DP ₆₇ DP ₇₁ DP ₇₃ DP ₇₈ DP ₈₁ DP ₈₂ DP ₈₇ DP ₈₉ DP ₉₂ DP ₉₃ DP ₉₅ DP ₉₇ DP ₉₉ DP ₁₀₀ DP ₁₀₈ DP ₁₁₄ DP ₁₂₂ DP ₁₃₂ DP ₁₃₅ DP ₁₃₉ DP ₁₆₂ DP ₁₇₀ DP ₁₇₃ DP ₁₇₈ DP ₁₇₉ DP ₁₉₃ DP ₂₁₄ DP ₂₂₆ DP ₂₃₈ DP ₂₄₄ DP ₂₅₆ DP ₂₅₇ DP ₂₆₆ DP ₂₆₇ DP ₂₇₀ DP ₂₇₃ DP ₂₇₇ DP ₂₇₈ DP ₂₇₉ DP ₂₈₅ DP ₂₈₆ DP ₂₉₀ DP ₂₉₃ DP ₂₉₈ DP ₃₀₄ DP ₃₀₈ DP ₃₂₁ DP ₃₂₆ DP ₃₂₇ DP ₃₃₀ DP ₃₃₁ DP ₃₃₄ DP ₃₃₈ DP ₃₃₉ DP ₃₅₂ DP ₃₅₃ DP ₃₆₂ DP ₃₆₄ DP ₃₆₈ DP ₃₆₉ DP ₃₇₂ DP ₃₇₄ DP ₃₇₅ DP ₃₇₆ DP ₃₇₈ DP ₃₇₉ DP ₃₈₀ DP ₃₈₄ DP ₃₉₁ DP ₃₉₈ DP ₃₉₉
F_{corr}	helix	CV ₁₀ RG ₃ EG ₅ CG ₁₀ OG ₁
	strand	CV ₁₇ MV ₁₆ MV ₂₆ HG ₂ CG ₁₀
S_{corr}	helix	CV ₁ CV ₅ CV ₁₀ CV ₁₁ CV ₁₃ CV ₁₆ CV ₁₇ MV ₁₁ MV ₁₀ MV ₁₆ MV ₂₁ MV ₂₁₀ MV ₂₁₆ RG ₁ RG ₃ XG ₁ XG ₃ HG ₂ EG ₂ EG ₅ CG ₃ CG ₄ CG ₅ CG ₆ CG ₉ CG ₁₀ OG ₁ DP ₁ DP ₉ DP ₁₀ DP ₄₁ DP ₆₁ DP ₇₀ DP ₁₁₆ DP ₁₈₁ DP ₁₈₉ DP ₁₉₅ DP ₂₁₀ DP ₃₀₆ DP ₃₁₆
	strand	CV ₁ CV ₁₀ CV ₁₆ CV ₁₇ CV ₁₈ MV ₁₆ MV ₁₇ MV ₂₁₆ MV ₂₁₇ RG ₃ RG ₄ XG ₁ HG ₂ EG ₅ CG ₄ CG ₁₀ OG ₁ OG ₄ DP ₉ DP ₁₀ DP ₃₀₆ DP ₃₁₆ DP ₃₁₈ DP ₃₃₈ DP ₃₅₇

Results achieved for subset S_{corr} for strand prediction are better than results of both AVDM and MLR methods, while the existing methods are better in case of helix content prediction, see Table 9. The AVDM method uses more advanced predictive model called analytic vector decomposition technique [5]. The MLR method uses MLR method, as in [8], but tests on the set of all 210 proteins. We anticipate that using more advanced prediction model in combination with feature selection performed in this paper would result in a system that surpasses the existing approaches.

4 Summary and Future Work

The paper presents a novel method for prediction of protein secondary structure content. The method is the first to consider alternative feature representation of primary protein sequences. It performs feature selection task to generate optimal, in terms of trade-off between prediction error rates and number of features, feature representation and performs MLR based prediction of the helix and strand protein content. The results based on the leave-one-out test for non-homologous protein sets show that not only 5-25 features set can be used to predict the secondary content values, but that the representation based only on 5 features can reduce error rates by 10% when compared to standard 20 features representation based on composition vector. The results for a comprehensive set of 6000 mixed homologous and non-homologous proteins also show that error rate reduction of 14-15% can be achieved when the proposed feature representation is used instead of standard composition vector based representation.

Future work will design 2-layer prediction system. First, protein structural class (α , β , and $\alpha\beta$) will be predicted and next specialized prediction models for each class and predicted structure will be used. Design is similar to [17] [18] [19] [20], but considers that structural class will be predicted, not assumed, and utilizes feature selection.

References

- [1] Berman H.M., et al.: The Protein Data Bank, *Nucleic Acids Research*, 28, 235-242, 2000
- [2] Bussian B., & Sender, C., How to Determine Protein Secondary Structure in Solution by Raman Spectroscopy: Practical Guide and Test Case DNsae I, *Biochem.*, 28, 4271-77, 1989
- [3] Boeckmann B., et al., The SWISS-PROT Protein Knowledgebase and Its Supplement TrEMBL in 2003, *Nucleic Acids Research*, 31, 365-370, 2003
- [4] Dwyer D., Electronic Properties of Amino Acids Side Chains Contribute to the Structural Preferences in Protein Folding, *J Bimolecular Structure & Dynamics*, 18:6, 881-892, 2001
- [5] Eisenhaber F., et al., Prediction of Secondary Structural Contents of Proteins from Their Amino Acid Composition Alone, I. New Analytic Vector Decomposition Methods, *Proteins*, 25:2, 157-168, 1996
- [6] Ganapathiraju M.K., et al., Characterization of Protein Secondary Structure, *IEEE Signal Processing Magazine*, 78-87, May 2004
- [7] Hobohm U., & Sander C., A Sequence Property Approach to Searching Protein Databases, *J. of Molecular Biology*, 251, 390-399, 1995

- [8] Krigbaum W., & Knutton S., Prediction of the Amount of Secondary Structure in a Globular Protein from its Amino Acid Composition, *Proc. of the Nat. Academy of Science*, 70, 2809-2813, 1973
- [9] Lodish H., et al., *Molecular Cell Biology*, 4th ed., W.H. Freeman & Company, New York, 50-54, 2000
- [10] Muskal S.M., & Kim S-H., Predicting Protein Secondary Structure Content: a Tandem Neural Network Approach, *J. of Molecular Biology*, 225, 713-727, 1992
- [11] Nelson D. & Cox M., *Lehninger Principles of Biochemistry Amino*, Worth Publish., 2000
- [12] Ruan J. et al., Highly Accurate and Consistent Method for Prediction of Helix and Strand Content from Primary Protein Sequences, *Artificial Intelligence in Medicine*, special issue on *Computational Intelligence Techniques in Bioinformatics*, accepted, 2005
- [13] Sreerama N., & Woody, R.W., Protein Secondary Structure from Circular Dichroism Spectroscopy, *J Molecular Biology*, 242, 497-507, 1994
- [14] Syed U., & Yona G., Using a Mixture of Probabilistic Decision Trees for Direct Prediction of Protein Function, *Proc. of RECOMB 2003 Conf.*, 224-234, 2003
- [15] Wang, J., et al., Application of Neural Networks to Biological Data Mining: a Case Study in Protein Sequence Classification, *Proc. of 6th ACM SIGKDD Inter. Conf. on Knowledge Discovery and Data Mining*, 305-309, 2000
- [16] Yang, X., & Wang, B., Weave Amino Acid Sequences for Protein Secondary Structure Prediction, *Proc. of 8th ACM SIGMOD workshop on Research issues in Data Mining and Knowledge Discovery*, 80-87, 2003
- [17] Zhang, C.T., Zhang, Z., & He. Z., Prediction of the Secondary Structure of Globular Proteins Based on Structural Classes, *J. of Protein Chemistry*, 15, 775-786, 1996
- [18] Zhang, C.T., et al., Prediction of Helix/Strand Content of Globular Proteins Based on Their Primary Sequences, *Protein Engineering*, 11:11, 971-979, 1998a
- [19] Zhang C.T., Zhang Z., & He Z., Prediction of the Secondary Structure Contents of Globular Proteins based on Three Structural Classes, *J Protein Chemistry*, 17, 261-272, 1998b
- [20] Zhang Z.D., Sun Z.R., & Zhang C.T., A New Approach to Predict the Helix/Strand Content of Globular Proteins, *J Theoretical Biology*, 208, 65-78, 2001

Alternative Clustering by Utilizing Multi-objective Genetic Algorithm with Linked-List Based Chromosome Encoding

Jun Du¹, Emin Erkan Korkmaz³, Reda Alhajj^{1,2}, and Ken Barker¹

¹ Department of Computer Science, University of Calgary,
Calgary, Alberta, Canada

{jundu, alhajj, barker}@cpsc.ucalgary.ca

² Department of Computer Science, Global University,
Beirut, Lebanon

³ Department of Computer Engineering, Yeditepe University,
Kadikoy, Istanbul, Turkey
ekorkmaz@cse.yeditepe.edu.tr

Abstract. In this paper, we present a linked-list based encoding scheme for multiple objectives based genetic algorithm (GA) to identify clusters in a partition. Our approach obtains the optimal partitions for all the possible numbers of clusters in the *Pareto Optimal* set returned by a single genetic GA run. The performance of the proposed approach has been tested using two well-known data sets, namely *Iris* and *Ruspini*. The obtained results are promising and demonstrate the applicability and effectiveness of the proposed approach.

Keywords: clustering, genetic algorithms, linkage-encoding, k-means, multi-objective optimization.

1 Introduction

In this paper, a new scheme is proposed for encoding clustering solutions into chromosomes. The proposed representation forms a linked-list structure for objects in the same cluster. The genetic operators modify the chromosomes by altering the links. Also, we deal with the partitional clustering problem by using a multi-objective GA [15] to minimize *Total Within Cluster Variation (TWCV)*, together with the number of clusters. TWCV [17] is a measure which denotes the sum of the average distance of cluster elements to cluster center. If this measure is used as the sole objective in the search, GA will tend to reduce the size of the clusters and eventually will form clusters with single elements where the variation turns out to be zero. Hence, traditionally, a prior specified number of clusters is needed for GA based k-clustering approaches treating TWCV as the single objective function. The other objective (minimizing the number of clusters) effectively handles this.

The new representation proposed in this paper is able to encode the solution space in fixed-length chromosomes. It enables an efficient exploration of the

solution space. Together with the use of multi-objective GA, this approach can be extended to deal with the general clustering problem where the optimum number of clusters is unknown. The *Pareto Optimal* set [15], obtained at the end of the run provides solutions with the optimum TWCV for various potential numbers of clusters.

Two well-known data sets *Iris* [3] and *Ruspini* [22] have been used in the experiments to demonstrate the applicability, usefulness and effectiveness of the proposed approach. These data sets have been widely used as benchmark problems for testing different techniques, and their corresponding optimal clustering is known. Hence, it is easy to evaluate the performance of a clustering method by using these data sets.

The rest of the paper is organized as follows. The objective functions used are discussed in Section 2. A closer look at the proposed approach is presented in Section 3. The experimental results obtained on two well-known data sets are reported in Section 4. Section 5 is conclusions.

2 The Objective Functions

Many optimization problems are multi-objective by nature. The classical approach to such problems is to use a single objective function which is obtained by a linear combination (weighted sum) of multiple objectives. Another approach is to treat different objectives as different constraints and use thresholds and penalties during the search. However, the usage of weights and penalties has been clearly proved problematic in the domain of GA. In our approach, the *Niched Pareto Genetic Algorithm* described in [15] is used in order to minimize the following two objectives.

1. Total within cluster variation (TWCV), which has been effectively used for the k -clustering problem.
2. Number of clusters.

The formal definition of TWCV is given in [17] as follows. Let the clustering problem be partitioning n objects, each has d different properties, into k different groups. So, each object can be represented as a vector with dimension d , and the collection of these objects would be a matrix X , where entry x_{ij} denotes the j^{th} property of the i^{th} object. Then, another matrix W can be defined as:

$$w_{ik} = \begin{cases} 1, & \text{if } i^{\text{th}} \text{ pattern belongs to } k^{\text{th}} \text{ cluster.} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The following two properties will hold for the new matrix; $w_{ij} \in \{0, 1\}$ and $\sum_{k=1}^K w_{ij} = 1$, where K is the total number of clusters.

Let $c_k = \{c_{k1}, c_{k2}, \dots, c_{kd}\}$ denote the center of the k^{th} cluster, then

$$c_{kj} = \frac{\sum_{i=1}^n w_{ik} x_{ij}}{\sum_{i=1}^n w_{ik}}. \quad (2)$$

The within-cluster variation (WCV) of the k^{th} cluster can be defined as

$$S^{(k)}(W) = \sum_{i=1}^n w_{ik} \sum_{j=1}^d (x_{ij} - c_{kj})^2 \tag{3}$$

Lastly, TWCV is defined as

$$S(W) = \sum_{k=1}^K S^{(k)} = \sum_{k=1}^K \sum_{i=1}^n w_{ik} \sum_{j=1}^d (x_{ij} - c_{kj})^2 \tag{4}$$

3 A Closer Look at the Proposed Approach

The most straightforward and the most widely used encoding scheme is *Group Number Encoding* [16]. In this scheme, the value of each gene represents the membership of an object to one of the clusters. Let the set of objects to be clustered in k groups be $\mathcal{O} = \{o_1, o_2, \dots, o_n\}$. Since one gene is reserved for each object, the length of the chromosomes will be n . Let V be a function denoting the value of a gene in a chromosome. If $\mathcal{C} = \{g_1, g_2, \dots, g_n\}$ is a chromosome in the population, where $\forall g_i \in \mathcal{C}, 1 \leq V(g_i) \leq k$, then $V(g_i)$ will denote the cluster number for object o_i . Two objects, o_i and o_j will be in the same cluster if and only if $V(g_i) = V(g_j)$.

For example, the sample chromosome 2316736211 would encode the clustering solution where the first object is in cluster 2, the second in 3 and so on. However, it is possible to have multiple distinct chromosomes for the same solution with this encoding. In a clustering process, the naming or the ordering of the clusters is irrelevant. For instance, renaming cluster 2 to cluster 5 in chromosome 2316736211 creates a new chromosome 5316736511. However, both chromosomes are mapping to the same clustering solution. The drawbacks of this traditional encoding are presented in [8], and it is pointed out in [20] that this encoding is against the minimal redundancy principles set for encoding scheme design. The remedy proposed in [8] is to use a length variable encoding scheme. It reduces redundancy of chromosome population, but in the meantime it adds redundancy inside a chromosome; it needs more genes to encode a solution than traditional encoding. The other deficiency of the length variable encoding is that it cannot take advantage of conventional simple crossover and mutation operators. This gives advantage to the *Linear Linkage Encoding* presented in this section; it is a fixed length encoding scheme without any type of redundancy.

Under linkage encoding scheme, although each gene still stores an integer, the value of the gene no longer directly denotes the membership of an object but its fellowship - this is the fundamental difference between the group number encoding and the linkage encoding. Each gene is a link from an object to another object of the same cluster. Given n objects, any partition on them can be described as a chromosome of length n . Two objects are in the same group if either object can be directed to the other object via the links. Without any

constraint, the state of redundancy is just as bad as that of the group number encoding because the number of feasible chromosomes is still n^n .

Linear linkage encoding is a restricted linkage encoding. Let the n genes be indexed inside a chromosome from 1 to n . The value of each gene in LL chromosome denotes the index of a fellow gene where the objects that corresponding to these two genes would be in the same cluster. We can also treat the stored index as an out-link from a node, and if a gene stores its own index, it depicts an ending node. To qualify an unrestricted linkage chromosome as a valid linear linkage encoding chromosome, there are two constraints the chromosome must comply to:

1. The integer value stored in each gene is greater than or equal to its index but less than or equal to n .
2. No two genes in the chromosome have the same value with the exception that at most two genes can have the same integer value if the integer is the index of an ending node.

Formally, let the set of objects to be clustered be $\mathcal{O} = \{o_1, o_2, \dots, o_n\}$ and let $\mathcal{C} = \{g_1, g_2, \dots, g_n\}$ be a sample chromosome in the population. Assume V is a function that denotes the value of a gene and I is the function which returns its index. Then, the following two properties hold for the LL encoding.

$$\forall g_i \in \mathcal{C} [I(g_i) \leq V(g_i) \leq n]. \tag{5}$$

$$\begin{aligned} \forall g_i, g_j \in \mathcal{C} [V(g_i) = V(g_j) \implies (i = j) \\ \vee ((i > j) \wedge (V(g_i) = I(g_i))) \\ \vee ((i < j) \wedge (V(g_j) = I(g_j)))]. \end{aligned} \tag{6}$$

The boolean function ($\varphi : \mathcal{O}\mathcal{X}\mathcal{O} \mapsto \{True, False\}$), which would determine if two given objects are in the same cluster or not, can be recursively defined. If o_i and o_j are two objects where $i < j$, then

$$\varphi(o_i, o_j) = \begin{cases} [V(g_i) = I(g_j)] \vee \\ \exists g_k [(i < k < j) \wedge \varphi(o_i, o_k) \wedge V(g_k) = I(g_j)] \end{cases} \tag{7}$$

Linear linkage encoding gets its name because objects in a cluster construct a pseudo linear path with the only loop allowed being a self loop link to mark the last node. It can be represented by the *labeled oriented pseudo* (LOP) graph.

A LOP graph is a labeled directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}(\mathcal{G}) = \{v_1, v_2, \dots, v_n\}$. A composition of \mathcal{G} is a partition of $\mathcal{V}(\mathcal{G})$ into disjointed oriented pseudo path graphs $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m$ with the following properties:

1. Disjoint paths: $\bigcup_{i=1}^m \mathcal{V}(\mathcal{G}_i) = \mathcal{V}(\mathcal{G})$ and for $i \neq j, \mathcal{V}(\mathcal{G}_i) \cap \mathcal{V}(\mathcal{G}_j) = \emptyset$
2. Non-backward oriented edges: If there is an edge e directed from vertex v_l to v_k , then $l \leq k$.
3. Balanced connectivity:
 - (a) $|\mathcal{E}(\mathcal{G})| = |\mathcal{V}(\mathcal{G})|$
 - (b) Each \mathcal{G}_i must have only one ending node with a self referencing directed edge exists. The ending node has an indegree of 2 and an outdegree of 1.

- (c) Each \mathcal{G}_i must have only one starting node whose indegree is 0 and outdegree is 1.
- (d) All other $|\mathcal{V}(\mathcal{G}_i)| - 2$ vertexes in \mathcal{G}_i have both their indegree and outdegree equal to 1.

Theorem 1: Given a set of objects \mathcal{S} , there is one to one mapping between the chromosomes of LL encoding and the possible partition schemes.

In order to prove this theorem the following lemmas are used.

Lemma 1: Linear linkage encoding is an implementation of the LOP graph.

Lemma 2: Given a set of objects \mathcal{S} , there exists one and only one composition of LOP graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ for each partition scheme of \mathcal{S} , where $|\mathcal{V}| = |\mathcal{S}|$.

Note that, there is only one possible ascending order within clusters for a possible partition scheme. Thus, there exists only one composition of G for each partition. Reversely, a LOP graph represents only a single partition scheme by definition. Based on Lemmas 1 and 2, it can be claimed that LL encoding makes a one-to-one mapping between the chromosomes and clustering solutions.

Corollary: The number of chromosomes corresponding to all possible partition schemes is given by the n^{th} Bell number.

The number of ways a set of n elements can be partitioned into non-empty subsets is called a Bell number [4]. According to Theorem 1, there is one-to-one correspondence between the chromosomes of LL encoding and the possible partition schemes. Hence, the number of chromosomes in consideration would be denoted by the n^{th} Bell number $B(n)$, too. Compared to LL encoding scheme, traditional group number encoding demands GA to work in a solution space of $\frac{n^n}{B(n)}$ times larger. When n is 10, $\frac{n^n}{B(n)}$ is about 10^5 .

Although LL encoding keeps only fellowship in genes, it also implies the membership of each object. Since each cluster must have one starting node and one ending node, both nodes can be used to identify a cluster. In practice, ending node is treated as the membership identifier for clusters because it is easier to detect. Apparently, finding the membership of an object in LL encoding requires only linear time.

The initial population should include diverse chromosomes. It is intuitive to achieve this goal by generating random chromosomes, which means each gene in a chromosome is assigned an integer randomly selected from the range 1 to n , where n is the number of objects to be clustered. However, the chromosomes generated this way may violate the restrictions of linear linkage encoding. Based on the first LL encoding constraint, each integer should be between its index and the maximum integer index number, inclusive. Therefore, a chromosome generator for creating each gene based on this constraint would be a better choice for diversity. Note that, the chromosomes produced this way still would not be fully complied with the constraints laid for linear linkage encoding. Obviously, backward links are prevented with this generator, but multiple nodes can link to the same node, violating the second constraint.

Note that multiple links are allowed during the initialization process. Later, we will see backward links in a chromosome emerge in the process of the mutation operation. Therefore, a recovery process is needed after the constructors, and later other GA operators are employed to rectify a chromosome into its legitimate format. The Rectifying algorithm used for the recovery process involves two correction steps. First, backward links are eliminated from a chromosome. Then, multiple links to a node (except for the ending nodes) are replaced with one link in and one link out.

The selection process is very similar to that of Niche Pareto GA described in [15]. A chromosome is said to be fitter or to dominate another one when it is superior to the latter in terms of all the objective functions used. If only a part of the objective values of one chromosome are better than the other's, neither chromosome is deemed dominant to the other. A chromosome can be compared to a set of chromosomes. It is dominated by the set if any individual in the set is fitter than it. Otherwise, the chromosome is not dominated by the set.

When two randomly selected chromosomes competing for a spot in the parent pool, they are not directly compared with each other. Rather, each is compared to a comparison set of chromosomes sampled from the current generation. If one of the competing chromosomes, say A , is dominated by the comparison set and the other chromosome, say B , is not dominated, then B advances to the parent pool. However, when both A and B are either dominated or not dominated by the set, the niche count of each chromosome is compared. The chromosome with the smaller niche count gets advantage. Niche count is an indicator of the solution density around a chromosome in a certain solution population. This approach encourages even distribution of solutions in the GA population [15].

In each generation, the Pareto dominant set is achieved through a search in the whole population. Every individual is compared with the rest. If a chromosome is not dominated by the rest, it is copied to the Pareto dominant set. The Pareto dominant set of the last generation contains the optimal solution.

In our experiments, one point crossover is adapted. The operation both allows different clusters to exchange partial contents and may split a cluster into two.

The classical mutation was implemented for LL-encoding and the test results were not encouraging. With the classical mutation, the out-link of a node is very likely to change to a different one, but the new out-link might still point to the same cluster. This results in no change in the chromosome after being rectified, and hence lessens the affect of the mutation during the search. The solution is to make sure that the mutated gene gains a link to a different cluster instead of just a different node. Also, when the mutated gene is again assigned to its original cluster, that cluster is split into two. Hence, the mutation might have two different effects on the chromosome; a sub-group of objects can be moved to a new cluster or a cluster can be split into two. This new mutation method changes the membership of set of objects rather than just a single object.

4 Experimental Results

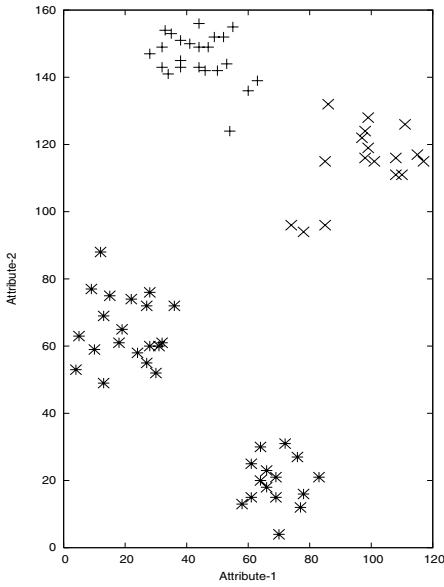
In this section, we report the result of our experiments on two widely used data sets, namely Iris and Ruspini. The former is a real dataset recording 150 observations of the three species (Setosa, Versicolor, and Virginica) of iris flowers, all described in four features. The latter is an artificial dataset of two attributes. Its 75 data points naturally form 4 clusters. Originally data points in both datasets are sorted so that the clusters are easily perceived. To obtain unbiased result, we reorganized the datasets and all the data points are randomly placed. In addition, data standardization process is applied to the dataset to neutralize the effects brought by data attributes of disparage magnitude. We divide each data element by the maximum value of its dimension to scale all data elements ranging from 0 to 1. In our experiments, two GAs are developed. Apart from the encoding schemes, all GA operators are kept the same as described in Section 3. The genetic parameters are fixed for all GA runs and they are presented in Table 1.

Note that the multi-objective GA tries to minimize TWCV for all possible number of clusters. For the Iris data set, the possible number of clusters ranges from 1 to 150. The single cluster is the case where all instances are placed into the same cluster, and each instance is considered as a separate cluster when number of clusters increases to 150. This range is between 1 and 75 for the Ruspini dataset since the number of instances is 75 in this domain. Note that the optimal number of clusters for Iris and Ruspini is 3 and 4, respectively. Hence, TWCV values obtained for smaller number of clusters is of more interest.

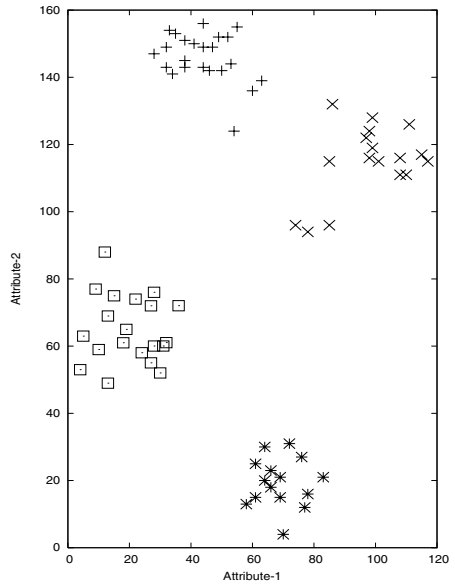
For Iris, the change in TWCV is quite stable down to 3 clusters. However, there is a considerable leap between TWCV values of 2 and 3 clusters. The same is valid for Ruspini between 3 and 4 clusters. Hence, it is possible to derive conclusions about the optimum number of clusters by considering the pareto optimal set obtained at the end of the GA search. In both data sets the optimum clusters are well separated from others. We realize that in a domain where the cluster borders are not very clear, the leap at the optimum number of clusters may not be as clear as the result obtained in these two domains.

Table 1. Genetic parameters used for the experiments

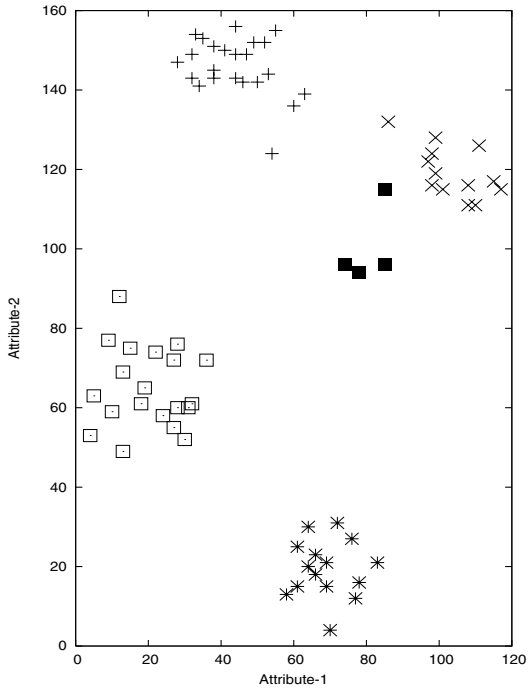
Parameter	Value
Number of Experiments	10
Number of Generations	2000
population size (Iris)	800
population size (Ruspini)	500
Niche Comparison Size (selection)	5
Nitch Radius	5
Nitch Count Size	25
Crossover Rate	0.9
Mutation Rate	0.2



(a)



(b)



(c)

Fig. 1. The best TWCV values obtained when the number of clusters is: a) 3; b) 4; c) 5

Note that Ruspini dataset has only two attributes. It is easy to display the clusters obtained on this data set as two-dimensional diagrams. In Figure 1 the best partitions obtained by the *Linkage encoding* are presented for 3, 4 and 5 clusters. From Figure 1, it can be easily realized that the natural partition of the dataset is formed by four clusters (Figure 1-b). However, the partitions obtained when the number of clusters is decreased to 3 or increased to 5 are also plausible. When the number of clusters is 3, two of the clusters that appear in the natural solution merge into a single cluster (Figure 1-a). On the other hand, one of the clusters in the natural solution splits into two when the number of clusters is 5. In this case, a new cluster is created with the elements that seem to be a bit more separate compared to the other elements in the original class (Figure 1-c).

5 Conclusions

In this paper, a new encoding scheme is proposed for the application of GA to the clustering problem. This new scheme has been successfully used with the multi-objective GA which is a powerful optimization technique. The results obtained on two well-known data sets provide a good insight about the importance and effectiveness of the new scheme. The analysis carried out clearly notifies that the new scheme is applicable, useful and effective. Although some extra processes are needed in order to keep the redundancy low, it has been observed that the computational cost of these processes is not significant.

The leap in TWCV after the optimum number of clusters seems to be an important issue about the proposed technique. The experiments demonstrate that it is expected to observe such a leap for datasets with well separated clusters. It would be interesting to observe the change in TWCV, in domains where cluster borders are not clear. In such domains, probably it would not be possible to directly observe the optimum number of clusters. However, an automatic analysis of the change in TWCV values might be helpful to determine the optimum point.

References

1. a Pen, J. Lozano, and J. Larran. An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recognition Letters*, 20(10):1027–1040, 1999.
2. Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.
3. C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 2000.
4. Andrew Bremner. Reviews: *The Book of Numbers*, by John Horton Conway and Richard K. Guy. *American Mathematical Monthly*, 104(9):884–??, November 1997.
5. Donald S. Burke, Kenneth A. De Jong, John J. Grefenstette, Connie Loggia Ramsey, and Annie S. Wu. Putting more genetics into genetic algorithms. *Evolutionary Computation*, 6(4):387–410, Winter 1998.
6. Rowena Marie Cole. Clustering with genetic algorithms. Master's thesis, Nedlands 6907, Australia, 1998.

7. I. Dhillon, J. Fan, and Y. Guan. Efficient clustering of very large document collections. In R. Grossman, C. Kamath, and R. Naburu, editors, *Data Mining for Scientific and Engineering Applications*. Kluwer Academic Publishers, 2001.
8. E. Falkenauer. *Genetic Algorithms and Grouping Problems*. John Wiley&Sons, 1998.
9. Emanuel Falkenauer. A new representation and operators for genetic algorithms applied to grouping problems. *Evolutionary Computation*, 2(2):123–144, 1994.
10. David Gibson, Jon M. Kleinberg, and Prabhakar Raghavan. Clustering categorical data: An approach based on dynamical systems. *VLDB Journal: Very Large Data Bases*, 8(3–4):222–236, February 2000.
11. D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
12. Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: an efficient clustering algorithm for large databases. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, Seattle, Washington, USA*, volume 27(2), pages 73–84, 1998.
13. J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.
14. John Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
15. Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg. A Niche Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, 1994.
16. Donald A. Jones and Mark A. Beltramo. Solving partitioning problems with genetic algorithms. In Lashon B. Belew, Richard K.; Booker, editor, *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 442–449. Morgan Kaufmann, July 1991.
17. K. Krishna and M. Murty. Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics - PartB: Cybernetics*, 29(3):433–439, 1999.
18. U. Maulik and S. Bandyopadhyay. Genetic algorithm-based clustering technique. *Pattern Recognition*, 33:1455–1465, 2000.
19. R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. In *Proc. of 1994 Int'l Conf. on Very Large Data Bases (VLDB'94)*, pages 144–155, September 1994.
20. Nicholas J. Radcliffe. Forma analysis and random respectful recombination. In Lashon B. Belew, Richard K.; Booker, editor, *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 222–229. Morgan Kaufmann, July 1991.
21. J. T. Richardson, M. R. Palmer, G. Liepina, and M. Hilliard. Some guidelines for genetic algorithms with penalty functions. In *Proc. of the 3rd Int. Conf. on Genetic Algorithms*, pages ?–? Morgan Kaufman, 1989.
22. E.H. Ruspini. Numerical methods for fuzzy clustering. *Inform. Sci.*, 2(3):19–150, 1970.
23. I. Sarafis, A. M. S. Zalzal, and P. Trinder. A genetic rule-based data clustering toolkit. In *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002*, pages 1238–1243, 2002.

Embedding Time Series Data for Classification

Akira Hayashi¹, Yuko Mizuhara², and Nobuo Suematsu¹

¹ Faculty of Information Sciences, Hiroshima City University,
3-4-1 Ozuka-higashi, Asaminami-ku, Hiroshima 731-3194, Japan
akira@im.hiroshima-cu.ac.jp

² Panasonic AVC Networks, 1-5 Matsuo-Cho,
Kadoma, 571-8505, Japan

Abstract. We propose an approach to embed time series data in a vector space based on the distances obtained from Dynamic Time Warping (DTW), and to classify them in the embedded space. Under the problem setting in which both labeled data and unlabeled data are given beforehand, we consider three embeddings, embedding in a Euclidean space by MDS, embedding in a Pseudo-Euclidean space, and embedding in a Euclidean space by the Laplacian eigenmap technique.

We have found through analysis and experiment that the embedding by the Laplacian eigenmap method leads to the best classification result. Furthermore, the proposed approach with Laplacian eigenmap embedding shows better performance than k -nearest neighbor method.

1 Introduction

1.1 Classification of Time Series Data

With the development of information technology, recognition of time series data, such as gesture recognition, video retrieval, online handwriting recognition, is becoming more important. Here, we consider the following 2 class classification problem for time series data.

A set of n time series data, $\mathcal{X} = \{X_1, \dots, X_n\}$, is given, where, X_i ($1 \leq i \leq n$) is a sequence of feature vectors whose length is l_i $X_i = (\mathbf{x}_1^i, \dots, \mathbf{x}_{l_i}^i)$. First s of the time series data $\{X_i \mid 1 \leq i \leq s\}$, are labeled with a class label $y_i \in \{-1, +1\}$. The task is to estimate the labels of unlabeled data: $\{X_i \mid s + 1 \leq i \leq n\}$.

Time series data are much more difficult to deal with than vector data with a fixed dimension. Many of the classification methods for time series data use generative models such as Hidden Markov Models (HMMs) [1]. When the true models are estimated correctly, these methods are accurate. But they need lots of training data. Other classification methods such as k nearest neighbors are based on distances which are obtained from dynamic time warping (DTW) [1]. k nearest neighbors can express class boundaries which have complex shapes without assuming the form of probability densities, but they tend to be sensitive to noise in general.

1.2 Proposed Approach

We propose a distance based approach which can be summarized as follows.

1. Compute the distances between time series data from DTW.
2. Map by Φ the time series data into a vector space (a feature space) \mathcal{F} , such that the DTW distances are preserved in some sense.

$$\begin{aligned} \Phi : \mathcal{X} &\rightarrow \mathcal{F} \\ X_i &\mapsto \Phi(X_i) \end{aligned}$$

Project $\Phi(X_i)$ to a lower dimensional subspace, and obtain $\tilde{\Phi}(X_i)$.

3. Train a classifier in \mathcal{F} using the labeled data $\{(\tilde{\Phi}(X_i), y_i) \mid 1 \leq i \leq s\}$.
4. Classify the unlabeled data, $\{\tilde{\Phi}(X_i) \mid s + 1 \leq i \leq n\}$, using the classifier.

1.3 Embedding in a Vector Space

We consider three methods, i.e. three kinds of mapping Φ , for embedding time series data in a vector space. The first method is multidimensional scaling (MDS [2], which embeds data in a Euclidean space. The second method, which is an extension of the first one, embeds data in a pseudo Euclidean space [3, 4, 5]. The third method uses the technique known as manifold learning [6, 7], and embeds time series data in a Euclidean space.

We consider the three embedding methods from the following reasons. MDS (the first method) is popular as an embedding method using distances between data. We consider the embedding in a pseudo Euclidean space (in the second method), because dynamic time warping distances do not satisfy the triangle inequality relationship for (Euclidean) distances. For the third embedding method, we have chosen, from several manifold learning techniques, one which can be applied to non vector data. Generally speaking, manifold learning embeds data on a manifold in a low dimensional space such that the geodesic distances are preserved. Manifold learning is gaining more and more attentions recently as a nonlinear dimensionality reduction method.

In order to analyze the three embedding methods, we employ the theoretical framework of kernel PCA [8], which is an extension of principal component analysis (PCA).

1.4 Related Work

Shimodaira et al. [9] propose dynamic time alignment kernel for voice recognition, and report better classification accuracy than HMMs when the number of training data is small. Bahlmann et al.[10] propose GDTW kernel, which substitutes the distance term in the Gaussian kernel with DTW distance, and obtained classification accuracy comparable with that of HMMs for online handwritten characters. However, neither method can prove the positive definiteness of the corresponding kernel matrix which guarantees the existence of a feature space (a Hilbert space).

Graepel et al. [4] embed data in a pseudo Euclidean space on the basis of the similarity measures of pairs of data, and then classify the embedded data using SVM. They experimented with cat's cortex data and protein data, and obtained a favorable result when compared with k-nearest neighbors. Pekalska et al. [5] propose a similar method and report a good result in the experiment to classify offline handwritten characters / object shapes in binary images. But, neither of them consider the classification of time series data.

Belkin et al. [6, 7] propose Laplacian eigenmap method to embed data in a low dimensional vector space based on a similarity matrix. They have experimented with offline handwritten digit classification and report a better result than k-nearest neighbors. But they do not consider time series data, either.

1.5 Paper Organization

After briefly explaining DTW (Sec. 2) and kernel PCA (Sec. 3), we propose the embedding methods in Sec. 4, compare the distance between data before and after the embeddings in Sec. 5, and explain the classification including the classifier learning in Sec. 6. We report on the experiment to evaluate the proposed methods in Sec. 7. Sec. 8 concludes the paper.

2 DTW

The DTW distances $\{d^2(X_i, X_j) \mid 1 \leq i, j \leq n\}$ which we use in the paper are computed as follows ($\|\cdot\|$ is the Euclidean norm.)

1. Initialize: $g(0, 0) = 0$
2. Repeat: for $1 \leq t_i \leq l_i, 1 \leq t_j \leq l_j$

$$g(t_i, t_j) = \min \begin{cases} g(t_i - 1, t_j) + \|\mathbf{x}_{t_i}^i - \mathbf{x}_{t_j}^j\|^2 \\ g(t_i - 1, t_j - 1) + 2\|\mathbf{x}_{t_i}^i - \mathbf{x}_{t_j}^j\|^2 \\ g(t_i, t_j - 1) + \|\mathbf{x}_{t_i}^i - \mathbf{x}_{t_j}^j\|^2 \end{cases}$$

3. Finish: $d^2(X_i, X_j) = g(l_i, l_j)/(l_i + l_j)$

3 Kernel PCA

We explain kernel PCA by following [8]. Let $\mathcal{X} = \{X_1, X_2, \dots\}$ be a finite or an infinite set (which need not be a subset of a vector space), and let k be a positive definite kernel function defined on $\mathcal{X} \times \mathcal{X}$. Then, there exists a mapping to a Hilbert space, $\Phi: \mathcal{X} \rightarrow \mathcal{H}$, and for any $X, X' \in \mathcal{X}$, $k(X, X') = \langle \Phi(X), \Phi(X') \rangle$ holds, where $\langle \cdot, \cdot \rangle$ stands for the inner product in the Hilbert space \mathcal{H} .

Kernel PCA performs the principal component analysis of the set: $\{\Phi(X_1), \Phi(X_2), \dots, \Phi(X_n)\}$. Let \mathbf{C} be the covariance matrix: $\mathbf{C} = \frac{1}{n} \sum_i \Phi(X_i) \Phi(X_i)^T$, and let λ and \mathbf{v} be an eigenvalue and eigenvector of the covariance matrix: $\mathbf{C}\mathbf{v} = \lambda\mathbf{v}$. Then, it can be shown that an eigenvector whose eigenvalue is not 0

is in the subspace spanned by $\{\Phi(X_i) \mid 1 \leq i \leq n\}$, and hence can be expanded as $\mathbf{v} = \sum_{i=1}^n \alpha_i \Phi(X_i)$. The expansion coefficients, $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^T$, are obtained from the eigenvectors of the Kernel matrix $\mathbf{K} : \mathbf{K}_{ij} = k(X_i, X_j)$ as follows.

$$n\lambda\boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha} \quad (1)$$

As a consequence, it can be shown that m -th principal component of $\Phi(X_i)$ can be computed by the following formula:

$$\langle \mathbf{v}^m, \Phi(X_i) \rangle = \sqrt{n\lambda_m} \boldsymbol{\alpha}^m(i) \quad (1 \leq i \leq n) \quad (2)$$

where λ_m is an eigenvalue of the matrix $\mathbf{K}\boldsymbol{\alpha}^m(i)$ is the i -th element of the m -th eigenvector $\boldsymbol{\alpha}^m$. Note that we have assumed that $\frac{1}{n} \sum_{i=1}^n \Phi(X_i) = \mathbf{0}$ in computing the covariance matrix, \mathbf{C} , but this can be easily realized by *centering* the kernel matrix \mathbf{K} .

4 Embedding in a Vector Space

4.1 First Method: MDS

The first method uses MDS[2] to obtain a mapping $\Phi_1 : \mathcal{X} \rightarrow \Re^n$ such that the following holds.

$$\|\Phi_1(X_i) - \Phi_1(X_j)\|^2 = d^2(X_i, X_j) \quad (1 \leq i, j \leq n) \quad (3)$$

We abbreviate $\Phi_1(X_i)$ as \mathbf{z}_i in what follows. In order to centralize the kernel matrix, let $\bar{\mathbf{z}} = \frac{1}{n} \sum_i \mathbf{z}_i$, and consider the inner product: $k_1(X_i, X_j) = \langle \mathbf{z}_i - \bar{\mathbf{z}}, \mathbf{z}_j - \bar{\mathbf{z}} \rangle$. Using (3), we can obtain, after some manipulations, a formula to compute the kernel matrix from DTW distances.

$$\begin{aligned} k_1(X_i, X_j) = & -\frac{1}{2}d^2(X_i, X_j) + \frac{1}{2n} \sum_{l=1}^n d^2(X_i, X_l) + \frac{1}{2n} \sum_{l=1}^n d^2(X_j, X_l) \\ & - \frac{1}{2n^2} \sum_{l=1}^n \sum_{m=1}^n d^2(X_l, X_m) \end{aligned} \quad (4)$$

The kernel matrix $\mathbf{K} : \mathbf{K}_{ij} = k_1(X_i, X_j)$ is decomposed through eigenvalue analysis as follows.

$$\mathbf{K} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T \quad (5)$$

where $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$, $\lambda_1 \geq \dots \geq \lambda_n$ is a diagonal matrix of the eigenvalues, and $\mathbf{U} = [\mathbf{e}^1, \dots, \mathbf{e}^n]$ is a matrix of the eigenvectors.

When \mathbf{K} is semi-positive definite, let $\mathbf{Z} = \boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{U}^T$, and then $\mathbf{K} = \mathbf{Z}^T\mathbf{Z}$ holds. Hence, we can view the i -th column of \mathbf{Z} as $\mathbf{z}_i - \bar{\mathbf{z}}$. Translate the origin to the centroid, keep up-to the p -th principal components, we obtain the following.

$$\tilde{\mathbf{z}}_i = (\sqrt{\lambda_1}\mathbf{e}^1(i), \sqrt{\lambda_2}\mathbf{e}^2(i), \dots, \sqrt{\lambda_p}\mathbf{e}^p(i))^T \quad (1 \leq i \leq n) \quad (6)$$

Note that the above (6) and (2) are identical up to a constant (\sqrt{n}).

Unfortunately, DTW distances do not satisfy the triangle inequality, and the matrix \mathbf{K} computed from (4) is not necessarily semi-positive definite. Nevertheless, the first method embeds data in the Euclidean space, simply by neglecting negative eigenvalues / vectors.

4.2 Second Method: Embedding in a Pseudo Euclidean Space

Like the first method, \mathbf{K} is computed from DTW distances using (4), and is eigen-decomposed as in (5). While the first method embeds data by viewing \mathbf{K} as a matrix of inner products in a Euclidean space, the second method embeds data by viewing \mathbf{K} as a matrix of symmetric bilinear forms in $\mathfrak{R}^{(n^+, n^-)}$, a pseudo Euclidean space¹, without neglecting negative eigenvalues / vectors [3, 4, 5].

The concrete embedding procedure is as follows. Take the absolute value $\bar{\mathbf{A}}$ of the eigenvalue matrix \mathbf{A} in (5): $\bar{\mathbf{A}} = \text{diag}(|\lambda_1|, |\lambda_2|, \dots, |\lambda_n|)$. Let \mathbf{Z} be $\bar{\mathbf{A}}^{\frac{1}{2}} \mathbf{U}^T$, choose p eigenvalues / vectors whose absolute values are the largest. The coordinates in the embedded space will be as follows.

$$\tilde{z}_i = (\sqrt{|\lambda_{m_1}|} e^{m_1}(i), \dots, \sqrt{|\lambda_{m_p}|} e^{m_p}(i))^T \quad (1 \leq i \leq n) \quad (7)$$

We explain briefly about a pseudo Euclidean space². A pseudo Euclidean space $\mathfrak{R}^{(n^+, n^-)}$ is a vector space with the following bilinear form : $\langle \cdot, \cdot \rangle_M$, which corresponds to the inner product in a Euclidean space.

$$\langle \mathbf{z}, \mathbf{z}' \rangle_M = \mathbf{z}^T \mathbf{M} \mathbf{z}'$$

$$\mathbf{M} = \begin{pmatrix} \mathbf{I}_{n^+} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I}_{n^-} \end{pmatrix}_{n \times n}$$

M is called as the signature matrix of a pseudo Euclidean space. For a pseudo Euclidean space, one can define, from its bilinear form $\langle \cdot, \cdot \rangle_M$, a pseudo metric (distance): $\|\mathbf{z} - \mathbf{z}'\|_M^2 = (\mathbf{z} - \mathbf{z}')^T \mathbf{M} (\mathbf{z} - \mathbf{z}')$. The second method seeks a mapping $\Phi_2 : \mathcal{X} \rightarrow \mathfrak{R}^{(n^+, n^-)}$ which satisfies

$$\|\Phi_2(X_i) - \Phi_2(X_j)\|_M^2 = d^2(X_i, X_j) \quad (1 \leq i, j \leq n) \quad (8)$$

4.3 Third Method: The Laplacian Eigenmap

The third method embeds time series data by employing the Laplacian Eigenmap technique [6, 7]. The procedure is as follows.

¹ n^+, n^- in $\mathfrak{R}^{(n^+, n^-)}$ stands for the number of positive and negative eigenvalues of \mathbf{K} , respectively.

² In the literature on pattern recognition, a feature space generally means a Hilbert space. The second method, however, considers a pseudo Euclidean space also as a feature space.

1. Compute the similarity matrix \mathbf{W} from DTW distances.

$$\mathbf{W}_{ij} = \begin{cases} e^{-d^2(X_i, X_j)/t} & i \neq j \wedge d(X_i, X_j) < \epsilon \\ 0 & \text{if otherwise} \end{cases} \quad (9)$$

where $t (> 0)$ is a hyper parameter.

2. Compute the Laplacian Matrix \mathbf{L} .

$\mathbf{L} = \mathbf{D} - \mathbf{W}$, \mathbf{D} is a diagonal matrix such that $D_{ii} = \sum_{j=1}^n \mathbf{W}_{ij}$.

3. Solve a generalized eigenvalue problem: $\mathbf{L}\mathbf{e} = \lambda\mathbf{D}\mathbf{e}$, and compute p smallest eigenvectors $\mathbf{e}^1, \dots, \mathbf{e}^p$ ($\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_p$)³.

4. Compute the coordinates in the embedded space.

Let $\tilde{\mathbf{U}} = [\mathbf{e}^1, \mathbf{e}^2, \dots, \mathbf{e}^p]$, and $\tilde{\mathbf{Z}} = [\tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_n] = \mathbf{U}^T$, that is to say

$$\tilde{\mathbf{z}}_i = (\mathbf{e}^1(i), \mathbf{e}^2(i), \dots, \mathbf{e}^p(i))^T \quad (1 \leq i \leq n) \quad (10)$$

5 Distances Before and After the Embedding

We investigate how the distances between time series data change when they are embedded in vector spaces.

As we can see clearly from Eq. (3) and (8), the first and the second methods perform, so to speak, a *global* embedding which maintains both short DTW distances and long DTW distances equally.

Next, let us consider the meaning of the solution for the generalized eigenvalue problem: $\mathbf{L}\mathbf{e} = \lambda\mathbf{D}\mathbf{e}$ in the third method [6]. To begin with, seek a mapping from time series data to p dimensional space, $\Psi : \mathcal{X} \rightarrow \mathcal{R}^p$ ($X_i \mapsto \tilde{\mathbf{z}}_i$), such that $\sum_i \sum_j \|\tilde{\mathbf{z}}_i - \tilde{\mathbf{z}}_j\|^2 \mathbf{W}_{ij}$ will be minimized. In other words, we try to map those time series which are close to each other (in DTW distances) to nearby points in \mathcal{R}^p . Let $n \times p$ matrix $\tilde{\mathbf{U}}$ be such that $\tilde{\mathbf{U}}^T = [\tilde{\mathbf{z}}_1 \tilde{\mathbf{z}}_2 \dots \tilde{\mathbf{z}}_n]$, then it can be shown that the following holds⁴.

$$\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \|\tilde{\mathbf{z}}_i - \tilde{\mathbf{z}}_j\|^2 \mathbf{W}_{ij} = \text{tr}(\tilde{\mathbf{U}}^T \mathbf{L} \tilde{\mathbf{U}}) \quad (11)$$

Hence, we are left with the following minimization problem.

$$\tilde{\mathbf{U}}^* = \underset{\tilde{\mathbf{U}}^T \mathbf{D} \tilde{\mathbf{U}} = \mathbf{I}}{\text{argmin}} \text{tr}(\tilde{\mathbf{U}}^T \mathbf{L} \tilde{\mathbf{U}}) \quad (12)$$

It is well known that the above minimization problem is reduced to finding the p smallest eigenvalues / vectors for a generalized eigenvalue problem: $\mathbf{L}\mathbf{e} = \lambda\mathbf{D}\mathbf{e}$. Since the distances after the embedding are weighted according to the similarity

³ It has recently been pointed out [11] that $\mathbf{K}_L = \mathbf{L}^\dagger$, i.e., the pseudo inverse of the Laplacian matrix \mathbf{L} is the kernel matrix for the Laplacian eigenmap technique. Therefore, embedding using the eigenvectors with the smallest eigenvalues of \mathbf{L} is equivalent to kernel PCA for the kernel matrix: \mathbf{K}_L .

⁴ From (11), we can see that \mathbf{L} is semi-positive definite.

\mathbf{W}_{ij} in (11), the third method performs, so to speak, a *local* embedding which maintains short DTW distances, neglecting long distances.

6 Classification

6.1 Classifier Training

We consider here linear classifiers only for simplicity.

$$f(X) = \langle \mathbf{w}, \tilde{\Phi}(X) \rangle + b = \langle \mathbf{w}, \tilde{\mathbf{z}} \rangle + b \quad (13)$$

The term $\langle \mathbf{w}, \tilde{\Phi}(X) \rangle$ in the above should be $\langle \mathbf{w}, \tilde{\Phi}(X) \rangle_{\mathbf{M}}$ for the second method. However, by setting $\mathbf{w}' = \mathbf{M}\mathbf{w}$, and considering $\langle \mathbf{w}, \tilde{\mathbf{z}} \rangle_{\mathbf{M}} = \langle \mathbf{w}', \tilde{\mathbf{z}} \rangle$, pseudo Euclidean coordinates will be treated as Euclidean coordinates from now on [4].

Least Mean Square Error. Here, we seek a linear classifier which minimizes the mean square error for the labeled data: $\{(\tilde{\Phi}(X_i), y_i) \mid 1 \leq i \leq s\}$. In other words, the following error function in terms of \mathbf{w}, b should be minimized.

$$\text{Err}(\mathbf{w}, b) = \sum_{i=1}^s \{y_i - (\langle \mathbf{w}, \tilde{\mathbf{z}}_i \rangle + b)\}^2 \quad (14)$$

Maximal Margin. A hyperplane which has the maximal margin will be obtained by minimizing $\|\mathbf{w}\|^2$ under the following constraint [12]

$$y_i(\langle \mathbf{w}, \tilde{\mathbf{z}}_i \rangle + b) \geq 1, \quad i = 1, \dots, s \quad (15)$$

6.2 Classifying Unlabeled Data

Let $f(X) = \langle \mathbf{w}^*, \tilde{\mathbf{z}} \rangle + b^*$, and for $X_i (i > s)$,

$$y_i = \begin{cases} 1, & \text{if } f(X_i) \geq 0 \\ -1, & \text{if } f(X_i) < 0 \end{cases} \quad (16)$$

7 Experiments

7.1 Experiment 1

We compare the three embedding methods using Australian sign language (ASL) data [13]. The ASL data consists of 95 signs obtained from 5 subjects, each of which is a sequence of 9 dimensional feature vectors. We picked up two pairs, "sad" and "what", "go" and "please" among the 95 signs, and classified each pairs into two classes. Each sign class has 70 samples.

As to the third method, the similarity matrix was computed by using 8 nearest neighbors instead of ϵ distance neighbors. The value for t , $t = 10000$, was determined experimentally.

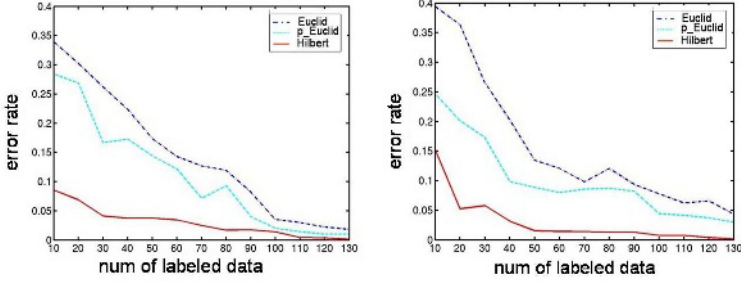


Fig. 1. Experiment 1 “sad” vs “what” (left) and “go” vs “please” (right). Average error rate from 30 trials are plotted. Solid lines (Hilbert) are for the third method

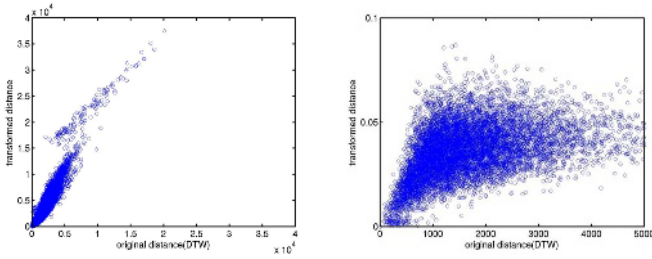


Fig. 2. Distances before and after the embedding: 1st Method (left) and 3rd Method (right) for 140 data of “go” and “please” signs

We varied the total number of training data. The rest were used as test data. The dimensionality of the embedded space, p , was set to 20% of the number of the training data. We used the maximal margin linear classifier for the first and second method, and the least mean squared error classifier for the third method, about which good results have been reported [4, 5, 7].

Fig. 1 shows that the third method has the best accuracy. Let us consider the reason. Since DTW distances are originally pattern matching scores, short distances which show a good match tend to be reliable, but long distances tend to be unreliable. While the first and the second method maintain both short and long distances equally alike in the embeddings, the third method tries to maintain only short distances by putting more weights on short distances. We thus conjecture that the third method has yielded the best result because it is compatible with the above nature of DTW distances. (See also Fig. 2, which supports the analysis in Sec. 5 on distances.)

7.2 Experiment 2

In the second experiment, we compare the third method: Laplacian eigenmap embedding, which had the highest accuracy in the first experiment, with k near-

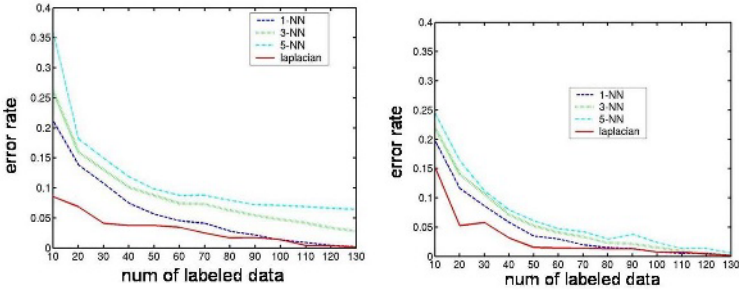


Fig. 3. Experiment 2: "sad" vs "what" (left) and "go" vs "please" (right). The average error rate is plotted for the same ASL data used in the first experiment

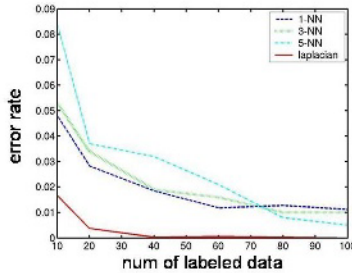


Fig. 4. Experiment 2: "upward shift" vs "increasing trend" in Control Chart Time Series [13]

est neighbors ($k = 1, 3, 5$). K nearest neighbors also use DTW distances, but do not embed data.

The results in Fig. 3 and 4 show that the third method has the best classification accuracy. We conjecture the reasons as follows. Firstly, the third method uses a linear classifier, which is expected to be more robust to noise than k nearest neighbors. Secondly, the third method seems to use unlabeled data effectively [14], because the coordinates of test data in the embedded space are determined by the DTW distances not only to the labeled data (the training data), but also to the unlabeled data (the test data), as far as they are within 8 nearest neighbors.

8 Conclusion

We have proposed an approach to embed time series data in a vector space based on the distances obtained by Dynamic Time Warping, and to classify them in the embedded space. Under the problem setting in which both labeled data and unlabeled data are given beforehand, we have considered three embeddings,

embedding in a Euclidean space by MDS, embedding in a Pseudo-Euclidean space, and embedding in a Euclidean space by the Laplacian eigenmap technique. We have found that embedding by the Laplacian eigenmap technique leads to the best classification result. Furthermore, the proposed approach with Laplacian eigenmap embedding shows better performance than k -nearest neighbors.

References

1. L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.
2. W.S. Torgerson, *Theory and methods of scaling*, J.Wiley & Sons, 1958.
3. L. Goldfarb, "A new approach to pattern recognition", *Progress in Pattern Recognition*, Elsevier Science Publishers B.V. vol.2, pp.241-402, 1985.
4. T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer, "Classification on pairwise proximity data", In *Advances in Neural Information Processing 11*, pp.438-444, 1999.
5. E. Pekalska, P. Paclik, and R.P.W Duin, "A generalized kernel approach to dissimilarity-based classification", *Journal of Machine Learning Research*, Special Issue on Kernel Methods, vol.2, no.2, pp.175-211, 2002.
6. M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering", *Advances in Neural Information Processing Systems 14*, pp.585-591, 2002.
7. M. Belkin and P. Niyogi, "Using manifold structure for partially labeled classification", *Advances in Neural Information Processing Systems 15*, pp.929-936, 2003.
8. B. Schölkopf, A.J. Smola, K.R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem", *Neural Computation*, 10, pp.1299-1319, 1998.
9. H. Shimodaira, K. Noma, M. Nakai, S. Sagayama, "Dynamic time-alignment kernel in support vector machine", *Neural Information Processing Systems 14*, pp.921-928, 2002.
10. C. Bahlmann, B. Haasdonk, and Hans Burkhardt, "On-line handwriting recognition with support vector machines-a kernel approach", *Proc. 8th Int. W/S on Frontiers in Handwriting Recognition*, pp.49-54, 2002.
11. J. Ham, D.D. Lee, S. Mika, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds", TR-110, Max-Planck-Institut für biologische Kybernetik, Tübingen, 2003.
12. C. Corres and V. Vapnik, "Support vector networks", *Machine Learning*, vol.20, pp.273-297, 1995.
13. S. Hettich and S.D. Bay, *UCI Repository of KDD Databases*, <http://kdd.ics.uci.edu/> 1999.
14. M. Seeger, "Learning with labeled and unlabeled data", Technical report, Institute for Adaptive and Neural Computation, University of Edinburgh, 2001.

Analysis of Time Series of Graphs: Prediction of Node Presence by Means of Decision Tree Learning

Horst Bunke¹, Peter Dickinson², Christophe Irmiger¹, and Miro Kraetzl²

¹ Department of Computer Science, University of Bern,
Neubrückstrasse 10, CH-3012 Bern, Switzerland
{bunke, irmiger}@iam.unibe.ch

² Intelligence, Surveillance and Reconnaissance Division,
Defence Science and Technology Organisation,
Edinburgh SA 5111, Australia
{Peter.Dickinson, Miro.Kraetzl}@dsto.defence.gov.au

Abstract. This paper is concerned with time series of graphs and proposes a novel scheme that is able to predict the presence or absence of nodes in a graph. The proposed scheme is based on decision trees that are induced from a training set of sample graphs. The work is motivated by applications in computer network monitoring. However, the proposed prediction method is generic and can be used in other applications as well. Experimental results with graphs derived from real computer networks indicate that a correct prediction rate of up to 97% can be achieved.

1 Introduction

Time series, or sequence, data are encountered in many applications, such as financial engineering, audio and video databases, biological and medical research, and weather forecast. Consequently, the analysis of time series has become an important area of research [1]. Particular attention has been paid to problems such as time series segmentation [2], retrieval of sequences or partial sequences [3], indexing [4], classification of time series [5], detection of frequent subsequences [6], periodicity detection [7] and prediction [8, 9, 10].

Typically a time series is given in terms of symbols, numbers, or vectors [1]. In the current paper we go one step further and consider time series of graphs. A time series of graphs is a sequence, $s = g_1, \dots, g_n$, where each g_i is a graph. In a recent survey it has been pointed out that graphs are a very suitable and powerful data structure for many operations needed in data mining in intelligent information processing [11]. As a matter of fact, traditional data structures, such as sequences of symbols, numbers, or vectors, can all be regarded as a special case of sequences of graphs.

The work presented in this paper is motivated by one particular application, which is computer network monitoring [12, 13]. In this application, graphs play

an important role [14]. The basic idea is to represent a computer network by a graph, where the clients and servers are modelled by nodes, and physical connections correspond to edges. If the state of the network is captured at regular points in time and represented as a graph, a sequence, or time series, of graphs is obtained that formally represents the network. Given such a sequence of graphs, abnormal network events can be detected by measuring the dissimilarity, or distance, between a pair of graphs that represent the network at two consecutive points in time. Typically an abnormal event manifests itself through a large graph distance [14].

In the current paper we address a different problem, viz. the recovery of incomplete network knowledge. Due to various reasons it may happen that the state of a network node or a network link can't be properly captured during network monitoring. This means that it is not known whether a certain node or edge is actually present or not in the graph sequence at a certain point in time. In this paper we describe a procedure that is able to recover missing information of this kind. This procedure is capable to make a decision as to the presence or absence of such a network node or edge. An information recovery procedure of this kind can also be used to predict, at time t , whether a certain computer or a certain link will be present, i.e. active, in the network at the next point in time, $t + 1$. Such procedures are useful in computer network monitoring in situations where one or more network probes have failed. Here the presence, or absence, of certain nodes and edges is not known. In these instances, the network management system would be unable to compute an accurate measurement of network change. The techniques described in this paper can be used to determine the likely status of this missing data and hence reduce false alarms of abnormal change.

Although the motivation of our work is in computer network monitoring, the methods described in this paper are fairly general and can be applied in other domains as well. Our proposed recovery scheme is based on decision tree learning as described in [15]. Basically we cast the recovery and prediction task into a classification framework, where one wants to decide whether a node is present in the network at a certain point in time or not. Clearly such a decision can be understood as a two-class classification problem, with one class, Ω_0 , indicating the absence of the node in question, and another class, Ω_1 , representing its presence in the network at a given point in time.

The rest of this paper is organized as follows. Basic terminology and notation will be introduced in the next section. Then, in Section 3, we will describe our novel information recovery and prediction scheme. Experimental results with this new scheme will be presented in Section 4 and conclusions drawn in Section 5.

2 Basic Concepts and Notation

A labeled graph is a 4-tuple, $g = (V, E, \alpha, \beta)$, where V is the finite set of nodes, $E \subseteq V \times V$ is the set of edges, $\alpha : V \rightarrow L$ is the node labeling function, and

$\beta : E \rightarrow L'$ is the edge labeling function, with L and L' being the set of node and edge labels, respectively. In this paper we focus our attention on a special class of graphs that are characterized by unique node labels. That is, for any two nodes, $x, y \in V$, if $x \neq y$ then $\alpha(x) \neq \alpha(y)$. Properties of this class of graphs have been studied in [16]. In particular it has been shown that problems such as graph isomorphism, subgraph isomorphism, maximum common subgraph, and graph edit distance computation can be solved in time that is only quadratic in the number of nodes of the larger of the two graphs involved.

To represent graphs with unique node labels in a convenient way, we drop set V and define each node in terms of its unique label. Hence a graph with unique node labels can be represented by a 3-tuple, $g = (L, E, \beta)$ where L is the set of node labels occurring in g , $E \subseteq L \times L$ is the set of edges, and $\beta : E \rightarrow L'$ is the edge labeling function [16]. The terms “node label” and “node” will be used synonymously in the remainder of this paper.

In this paper we will consider time series of graphs, i.e. graph sequences, $s = g_1, g_2, \dots, g_N$. The notation $g_i = (L_i, E_i, \beta_i)$ will be used to represent individual graph g_i in sequence s ; $i = 1, \dots, N$. Motivated by the computer network analysis application considered in this paper, we assume the existence of a universal set of node labels, or nodes, \mathcal{L} , from which all node labels that occur in a sequence s are drawn. That is, $L_i \subseteq \mathcal{L}$ for $i = 1, \dots, N$ and $\mathcal{L} = \bigcup_{i=1}^N L_i$.¹

Given a time series of graphs, $s = g_1, g_2, \dots, g_N$, and its corresponding universal set of node labels, \mathcal{L} , we can represent each graph, $g_i = (L_i, E_i, \beta_i)$, in this series as a 3-tuple $(\gamma_i, \delta_i, \hat{\beta}_i)$ where

- $\gamma_i : \mathcal{L} \rightarrow \{0, 1\}$ is a mapping that indicates whether node l is present in g_i or not. If l is present in g_i , then $\gamma_i(l) = 1$; otherwise $\gamma_i(l) = 0$.²
- $\delta_i : \mathcal{L}' \times \mathcal{L}' \rightarrow \{0, 1\}$ is a mapping that indicates whether edge (l_1, l_2) is present in g_i or not; here we choose $\mathcal{L}' = \{l \mid \gamma_i(l) = 1\}$, i.e. \mathcal{L}' is the set of nodes that are actually present in g_i .
- $\hat{\beta}_i : \mathcal{L}' \times \mathcal{L}' \rightarrow L'$ is a mapping that is defined as follows:

$$\hat{\beta}_i(e) = \begin{cases} \beta_i(e), & \text{if } e \in \{(l_1, l_2) \mid \delta_i(l_1, l_2) = 1\} \\ \text{undefined,} & \text{otherwise} \end{cases}$$

The definition of $\hat{\beta}_i(e)$ means that each edge e that is present in g_i will have label $\beta_i(e)$. The 3-tuple $(\gamma_i, \delta_i, \hat{\beta}_i)$ that is constructed from $g_i = (L_i, E_i, \beta_i)$ will be called the *characteristic representation* of g_i , and denoted by $\chi(g_i)$. Clearly, for any given graph sequence $s = g_1, g_2, \dots, g_N$ the corresponding sequence $\chi(s) = \chi(g_1), \chi(g_2), \dots, \chi(g_N)$ can be easily constructed and is uniquely defined. Conversely, given $\chi(s) = \chi(g_1), \chi(g_2), \dots, \chi(g_N)$ we can uniquely reconstruct $s = g_1, g_2, \dots, g_N$.

¹ In the computer network analysis application \mathcal{L} will be, for example, the set of all unique IP host addresses in the network. Note that in one particular graph, g_i , usually only a subset is actually present. In general, \mathcal{L} may be any finite or infinite set.

² One can easily verify that $\{l \mid \gamma_i(l) = 1\} = L_i$.

In the current paper we'll pay particular attention to graph sequences with missing information. There are two possible cases of interest. First it may not be known whether node l is present in graph g_i or not. In other words, in $\chi(g_i)$ it is not known whether $\gamma_i(l) = 1$ or $\gamma_i(l) = 0$. Secondly, it may not be known whether edge (l_1, l_2) is present in g_i , which is equivalent to not knowing, in $\chi(g_i)$, whether $\delta_i(l_1, l_2) = 1$ or $\delta_i(l_1, l_2) = 0$. In this paper, we focus on the case of missing node information. To cope with the problem of missing information and in order to make our notation more convenient, we extend function γ in the characteristic representation, $\chi(g)$, of graph $g = (L, E, \beta)$ by including the special symbol $?$ in the range of values of each function to indicate the case of missing information. That is, we write $\gamma(l) = ?$ if it is unknown whether node l is present in g or not.

3 Recovery of Missing Information Using Decision Trees

Our goal is to construct a function that computes $\gamma_t(l)$ for a node, l , given some data extracted from time series g_1, g_2, \dots, g_t as input. In the approach proposed in this paper the function that computes $\gamma_t(l)$ will actually use only information extracted from g_t . However, graphs g_1, \dots, g_{t-1} will be used as a training set, i.e. they are used to learn this function.

The approach proposed in this paper is based on decision trees. Decision tree classifiers have often been used for the purpose of object classification. An object, \mathbf{x} , is given in terms of the values of d different features and represented by means of a d -dimensional vector, i.e. $\mathbf{x} = (x_1, \dots, x_d)$. The feature values, x_i , $1 \leq i \leq d$, can be numerical or non-numerical. It is possible that one or several feature values are unknown. To classify an object means to assign it to a class, Ω_i , out of a number of given classes, $\Omega_1, \dots, \Omega_c$. For all further technical details we refer the reader to [15].

Assume we want to make a decision as to $\gamma_t(l) = 0$ or $\gamma_t(l) = 1$, given $\gamma_t(l) = ?$. Actually, this decision problem can be transformed into a classification problem as follows. The network at time t , g_t , corresponds to the unknown object to be classified. Network g_t is described by means of a feature vector, $\mathbf{x} = (x_1, \dots, x_d)$, and the decision as to $\gamma_t(l) = 0$ or $\gamma_t(l) = 1$ can be interpreted as a two-class classification problem, where $\gamma_t(l) = 0$ corresponds to class Ω_0 and $\gamma_t(l) = 1$ corresponds to class Ω_1 . As features x_1, \dots, x_d that represent the unknown object \mathbf{x} , i.e. graph g_t , one can use, in principle, any quantity that is extractable from graphs g_1, \dots, g_t . In this paper we consider the case where these features are extracted from graph g_t exclusively. Assume that the universal set of node labels is given by $\mathcal{L} = \{l_0, l_1, \dots, l_D\}$, and assume furthermore that it is node label l_0 for which we want to make a decision as to $\gamma_t(l_0) = 0$ or $\gamma_t(l_0) = 1$, given $\gamma_t(l_0) = ?$. Then we set $d = D$ and use the D -dimensional binary feature vector $(\gamma_t(l_1), \dots, \gamma_t(l_D))$ to represent graph g_t . In other words, $\mathbf{x} = (\gamma_t(l_1), \dots, \gamma_t(l_D))$. This feature vector is to be classified as either belonging to class Ω_0 or Ω_1 . The former case correspond to deciding $\gamma_t(l_0) = 0$, and the latter to $\gamma_t(l_0) = 1$. Intuitively, using $(\gamma_t(l_1), \dots, \gamma_t(l_D))$ as a feature vector for

the classification of g_t means we make a decision as to the presence or absence of l_0 in g_t depending on the presence or absence of all other nodes from \mathcal{L} in g_t .³

For the implementation of the classification procedure described in the last paragraph, we need a training set. For the training set we can use all previous graphs in the given time series, i.e. g_1, \dots, g_{t-1} . From each graph, g_i , we extract the D -dimensional feature vector

$$\mathbf{x}_i = (\gamma_i(l_1), \dots, \gamma_i(l_D)) \quad (3.1)$$

So our training set becomes $L = \{\mathbf{x}_1, \dots, \mathbf{x}_{t-1}\}$. We do need to assign the proper class to each element of the training set. This can be easily accomplished by assigning class Ω_0 to \mathbf{x}_i if $\gamma_i(l_0) = 0$; otherwise, if $\gamma_i(l_0) = 1$, we assign class Ω_1 to \mathbf{x}_i ; $i = 1, \dots, t-1$.

Given such a training set, constructed from g_1, \dots, g_{t-1} , we can now apply any known procedure to infer a decision tree from training set L . In the experiments described in Section 4, we have used C4.5 [15]. Once the decision tree has been produced, it is straightforward to classify feature vector \mathbf{x}_t (see Eq. (3.1)), which describes g_t , as belonging to Ω_0 or Ω_1 .

Decision tree classifiers are able to deal with unknown attribute values. This is important in our application because we must expect that not only information about node l_0 in g_t is missing, but also about other nodes, l_i , in g_t , where $i \in \{1, \dots, D\}$. Similarly, when building the decision tree from training set $L = \{\mathbf{x}_1, \dots, \mathbf{x}_{t-1}\}$, there may be graphs, g_i , $i \in \{1, \dots, t-1\}$ where it is not known for some nodes whether they are present or not in g_i . Hence some of the $\gamma_i(l_j)$ may be unknown. Fortunately, decision tree induction methods are able to cope with such cases of missing data [15].

The procedure described in this section is based on two assumptions. The first assumption is that there is some kind of correlation between the occurrence of a node, l , in graph g_t , and the occurrence of some (or all) other nodes in the same graph. In other words, we assume that the behaviour of node l is dependent, in some way, on the behaviour of the other nodes. Note, however, that we don't need to make any assumptions as to the mathematical nature of this dependency. Our second assumption is that there is some stationarity in the dependency between l and the other nodes. Using graphs g_1, \dots, g_{t-1} as a training set to derive a classifier that makes a decision pertaining to graph g_t will work well only if the dependency between l and the other nodes in g_t is of the same nature as in g_1, \dots, g_{t-1} .

In a practical setting it may be computationally too demanding to infer a decision tree at each point of time, t . Hence it may be preferable to do an update of the actual decision tree only after a certain period of time has elapsed. Moreover, in the decision tree updating process it is possible to use only part of the network history. This means that for the construction of the decision tree for g_t , we don't use g_1, \dots, g_{t-1} , but focus on only the M most recent

³ Note that in principle also information about edges could be incorporated in the feature vector. However such an extension would increase the space complexity from $O(D)$ to $O(D^2)$.

Table 1. Characterisation of the graph sequences used in the experiments

	S_1	S_2	S_3	S_4
Number of graphs in sequence	102	292	202	99
Size of smallest graph in sequence	38	85	15	572
Size of largest graph in sequence	94	154	329	10704
Average size of graphs in sequence	69.7	118.5	103.9	5657.8

graphs g_{t-M}, \dots, g_{t-1} . This is particularly advisable if there is evidence that the behaviour of the network is not perfectly stationary, but changing over time.

4 Experimental Results

The method described in Section 3 of this paper has been implemented and experimentally evaluated on real network data. For the experiments four time series of graphs, S_1 , S_2 , S_3 and S_4 , acquired from existing computer networks have been used. Characteristics of these graph sequences are shown in Table 1, where the size of a graph is defined as the number of its nodes. All four series represent logical communications on the network. Series S_1 , S_2 and S_4 were derived from data collected from a large enterprise data network, while S_3 was collected from a wireless LAN used by delegates during the World Congress for Information Technology (WCIT2002). The nodes in each graph of S_1 and S_2 represent business domains in the network, while in S_3 and S_4 they represent individual IP addresses. Note that all graph sequences are complete, i.e. there are no missing nodes and edges in these sequences.

For the experiments described in this section each time series is divided into two disjoint sets of graphs. The first set, G_1 , consists of all graphs g_i with index i being an odd number (i.e. graphs g_1, g_3, g_5, \dots), while the other set, G_2 , includes all graphs with an even index i (i.e. graphs g_2, g_4, \dots). First, set G_1 is used as the training set for decision tree induction and G_2 serves as the test set. Then G_1 and G_2 change their role, i.e. G_1 becomes the test and G_2 the training set. In the learning phase an individual decision tree is build for each node, i.e. for each label, l , belonging to the universal set of node labels, \mathcal{L} , as described in Section 3. Once all decision trees have been learned, testing takes place by assuming, for each graph g_i from the test set and each node label $l \in \mathcal{L}$, that $\gamma_i(l) = ?$. The decision tree learned for label l is used to decide either $\gamma_i(l) = 0$ or $\gamma_i(l) = 1$. Then the predicted value is compared to the real value. For each graph, g_i , in the test set we count the number of nodes that have been correctly predicted and divide this number by the total number of nodes in g . Splitting the considered time series of graph, S , into disjoint sets, G_1 and G_2 , is equivalent to performing a two-fold cross-validation, where each graph in time series S serves one time as a training and one time as a test sample. Clearly, a number of alternative scenarios for testing the proposed method are feasible. For example, instead of performing just a two-fold cross-validation, one could split the dataset into $n > 2$ disjoint subsets and do an n -fold cross-validation.

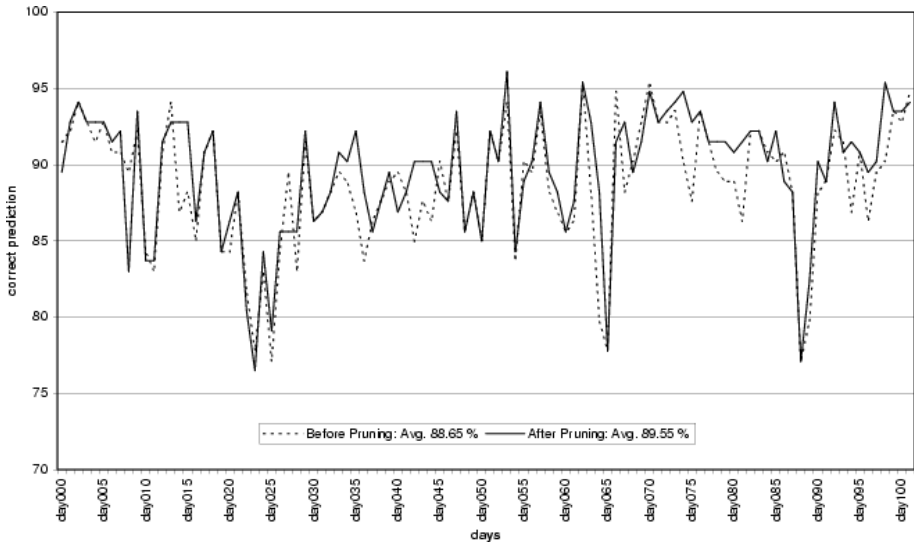


Fig. 1. Percentage of correctly predicted nodes in sequence S_1 , using both pruned and non-pruned decision trees

Fig. 1 shows the correct prediction rate for each graph in time series S_1 . Two different versions of the decision tree induction procedure were applied. The first version generates a tree without any pruning, while the second version applies pruning, as described in [15].

For each of the two versions the percentage of correctly predicted nodes in the corresponding graph is shown. We observe that the correct prediction rate is around 89% on the average. This is a remarkably high value taking into consideration that for a two-class classification problem, such as the one considered here, random guessing would give us an expected performance of only 50%. There are some obvious drops in prediction performance in Fig. 1, for example between time 20 and time 25, and around time 65. These drops correspond to abnormal events in the underlying computer network where major changes in network topology take place. Obviously these abnormal events don't follow the normal network behaviour represented in the training set. But the correct prediction rate is still quite high (more than 75% in any case).

From Fig. 1 it is hard to tell which of the two decision tree induction methods, including or excluding pruning, gives the better overall result. However if we average the correct prediction rate over all graphs in the time series, we get values of 89,5% and 88,6% for pruned and non-pruned trees, respectively. Hence using decision tree pruning gives us a slightly better performance.

Fig. 2 shows the percentage of correctly predicted nodes in sequence S_2 for pruned and non-pruned trees. We observe again a high correct prediction rate, with the curve showing a bit more jitter than Fig. 1. The correct prediction rate, averaged over all graphs of Sequence S_2 , is 93,4 for pruned and 92,8 for non-pruned decision trees.

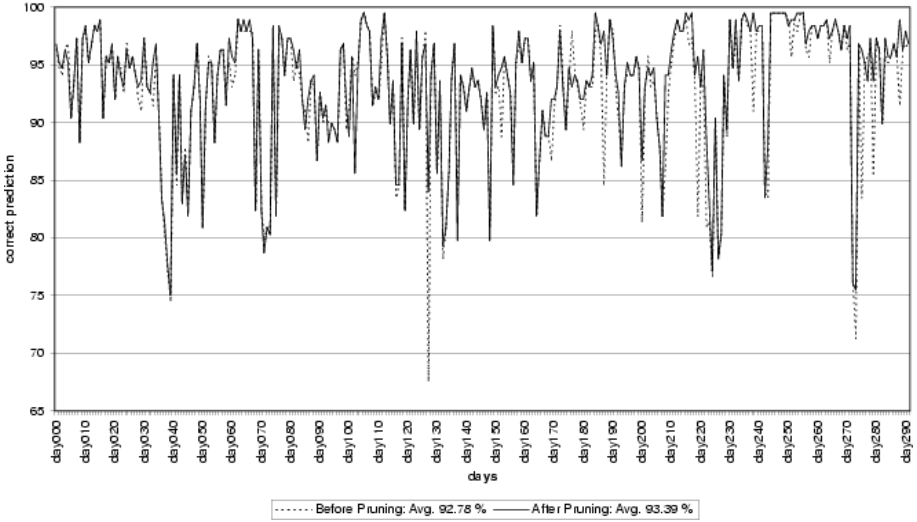


Fig. 2. Percentage of correctly predicted nodes in sequence S_2 , using both pruned and non-pruned decision trees

Table 2. Summary of correct prediction rates for sequences S_1 to S_4

	S_1	S_2	S_3	S_4
Pruned trees	89,5	93,4	96,9	89,4
Non-pruned trees	88,6	92,8	96,3	87,4

Results for sequences S_3 and S_4 are similar. A summary of the correct prediction rates, averaged over all graphs in a sequence, for both pruned and non-pruned decision trees, is shown in Table 2. From this figure we can conclude that for all time series used in this study quite high prediction rates were achieved. Pruned trees are consistently slightly better than non-pruned trees.

5 Conclusions

The problem of incomplete knowledge recovery and prediction of the behaviour of nodes in time series of graphs is studied in this paper. Formally, this task is formulated as a classification problem where nodes with an unknown status are to be assigned to one of the classes 'present in' (Ω_1) or 'absent from' (Ω_0) the actual graph. A decision tree learning scheme is proposed in order to solve this classification problem. The motivation of this work derives from the field of computer network monitoring. However the proposed framework for graph sequence analysis is fairly general and can be applied in other domains as well. In computer network monitoring, prediction procedures, as studied in this paper, are important for patching missing network data in instances where one or more

network probes have failed. Without such procedures, the network management system would have diminished capability in detecting abnormal change.

The proposed prediction procedure is straightforward to implement using decision tree induction software tools. Excellent performance with correct prediction rates ranging between about 89% and 97% with pruned trees has been achieved, using the proposed method on four independent data sets acquired from real computer networks.

The task in this paper has been cast as a classification problem. Consequently, not only decision trees, but also any other type of classifier can be applied, for example, neural network, Bayes classifier, nearest neighbor, or support vector machine. However, decision trees have at least two advantages. First, they can cope with missing data in both the training and test set. Secondly, it is possible to extract, from a trained decision tree, a number of rules that are interpretable by a human expert. This second aspect has not been stressed in our work yet, but is interesting to be investigated in future research.

The proposed schemes can be extended in a variety of ways. First of all, a prediction scheme, similar to the one proposed in this paper for nodes, can be designed for edges. In both, node and edge prediction, node as well as edge information can be utilized. That is, feature vectors as described in Section 3 can be extended to including information about the presence or absence of edges and they can be used for both node and edge prediction. Dealing with edges, however, introduces a substantially higher cost from the computational complexity point of view, because in a graph with n nodes we may have up to $O(n^2)$ edges, i.e. the complexity of our algorithms is going up from $O(n)$ to $O(n^2)$.

In the scheme proposed in Section 3 we have used within-graph context exclusively, i.e. the only information that is used in order to make a decision as to the presence or absence of a certain node in a graph, g , comes from that graph, g . One could use, however, also context in time. This means that we include information about the past behaviour of a network node in order to decide about its presence in the actual graph. From the conceptual point of view it is straightforward to integrate information of this kind in the proposed decision tree learning procedures. A more detailed investigation of this issue is left to future research.

Acknowledgement

The authors are thankful to J. Marbach and T. Varga for valuable contributions to this paper.

References

1. Last, M., Kandel, A., Bunke, H. (eds.): *Data Mining in Time Series Databases*, World Scientific, 2004
2. Keogh, E. et al: Segmenting time series: A survey and novel approach, in [1], 1-21

3. Kahveci, T., Singh, K.: Optimizing similarity search for arbitrary length time series queries, *IEEE Trans. KDE*, Vol. 16, No 2, 2004, 418-433
4. Vlachos, M. et al.: Indexing time-series under conditions of noise, in [1], 67-100
5. Zeira, G. et al: Change detection in classification models induced from time series data, in [1], 101-125
6. Tanaka, H., Uehara, K.: Discover motifs in multi-dimensional time-series using the principle component analysis and the MDL principle, in Perner, P., Rosenfeld, A. (eds.): *Machine Learning and Data Mining in Pattern Recognition, Proc. 3rd Int. Conference*, Springer LNAI 2734, 2003, 252-265
7. Yang, J., Wang, W., Yu, P.S.: Mining asynchronous periodic patterns in time series data, *IEEE Trans. KDE*, Vol. 15, No 3, 2003, 613-628
8. Schmidt, R., Gierl, L.: Temporal abstractions and case-based reasoning for medical course data: two prognostic applications, in Perner, P. (ed.): *Machine Learning in Pattern Recognition Proc. 2nd Int. Workshop*, Springer, LNAI 2123, 2001, 23-34
9. Fung, G.P.C.F., Yu, D.X., Lam, W.: News sensitive stock trend prediction, in Chen, M.-S., Yu, P.S., Liu, B. (eds.): *Advances in Knowledge Discovery and Data Mining, Proc. 6th Pacific-Asia Conference, PAKDD*, Springer, LNAI 2336, 2002, 481-493
10. Povinelli, R.F., Feng, X.: A new temporal pattern identification method for characterization and prediction of complex time series events, *IEEE Trans. KDE*, Vol. 15, No 2, 2003, 339-352
11. Bunke, H.: Graph-based tools for data mining and machine learning, in Perner, P., Rosenfeld, A. (eds.): *Machine Learning and Data Mining in Pattern Recognition, Proc. 3rd Int. Conference*, Springer LNAI 2734, 2003, 7-19
12. Hayes, S.: Analysing network performance management, *IEEE Communications Magazine*, 31 (5):52-58, May 1993
13. Higginbottom, G.N.: *Performance Evaluation of Communication Networks*, Artech House, Massachusetts, 1998
14. Bunke, H., Kraetzel, M., Shoubridge, P., Wallis, W.: Detection of abnormal change in time series of graphs, *Journal of Interconnection Networks*, Vol. 3, Nos 1,2, 2002, 85-101
15. Quinlan, R.: *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publ., 1993
16. Dickinson, P., Bunke, H., Dadej, A., Kraetzel, M.: Matching graphs with unique node labels, accepted for publication in *Pattern Analysis and Applications*

Disjunctive Sequential Patterns on Single Data Sequence and Its Anti-monotonicity

Kazuhiro Shimizu and Takao Miura

Dept.of Elect.& Elect. Engr., HOSEI University,
3-7-2 KajinoCho, Koganei, Tokyo, 184-8584 Japan

Abstract. In this work, we propose a novel method for mining frequent disjunctive patterns on single data sequence. For this purpose, we introduce a sophisticated measure that satisfies *anti-monotonicity*, by which we can discuss efficient mining algorithm based on APRIORI. We discuss some experimental results.

Keywords: Disjunctive Sequence Pattern, Anti-Monotonicity, Single Data Sequence.

1 Motivation

Recently there have been much investigation discussed to text data analysis using data mining techniques proposed so far. This is because generally text data has been analyzed qualitatively but not quantitatively, which means it is hard to apply several quantitative techniques such as statistical tests and data mining approach[6, 9]. However, much attention has been focused on quantitative analysis of several features of text thus new approach, called *text mining*, has been proposed to extend traditional methodologies.

Main ideas in text mining approach come from *frequency* and *co-occurrence*, the former means important words arise many times while the latter says related words occur at the same time[12, 14]. For example, when we analyze purchase logs in some bookstore, we could obtain co-occurrence of “*Snakes and Earrings*” and “*The Back One Wants to Kick*” and arrange the books on shelf or promoting the campaign¹.

The idea can be extended to many activities. Here we discuss em sequence data that means an ordered list of information along with time or any other and we extract patterns from the list as common behavior. In our case, we could obtain correlation that says those who get *Snakes and Earrings* would also get *The Back One Wants to Kick* in a month, and we would arrange some promotion behavior such as direct-mail or any other campaign *in the month*. Also we could apply the techniques for analyzing Web access patterns, medical-care and DNA sequence[8, 10, 11].

¹ The both books win the 130-th *Akutagawa* award in 2003. The authors, Ms. Hitomi Kanehara and Ms. Risa Wataya, are in 20's and have gotten into the news in Japan.

Among others, text mining has been paid much attention recently where *text* describes some context consisting of sequence of words. The main idea is similar to data mining, i.e., *frequency* and *co-occurrence*. We extract frequent patterns (lists of words, phrase) and we summarize (abstract) and label important events from them.

Clearly it is hard to extract all the frequent patterns from very long lists of words, since the search-space is so huge that we face to combinatorial explosion to the problems. For instance, we might have some sentences:

- (1) I met a mother of my friend's brother. Next day, I met a mother of my brother's friend.
- (2) I saw a red and white flag². Next day, I saw a white and red flag.

Clearly, in (1), we have two distinct mothers while, in (2), we see a same flag. To extract frequent patterns, we should count separately in (1) but count double in (2). In text mining, we want to extract a pattern *[red, white] flag* that means a pattern contains both a *red white flag* and a *white red flag* by ignoring permutation of *red* and *white*. This is called a *disjunctive* pattern or a *permuted* pattern.

We might think about regular expression by adding Kleene closure but we avoid the extension because of vast amount of complexity in this work.

When we examine disjunctive patterns, we might generate huge number of candidate patterns such as "*Snakes and Earrings The Back One Wants to Kick*". To obtain smaller search space, we need to scan a database many times thus we should have heavy I/O access to the data on hard disks. Sampling or any specialized data storage techniques might be helpful but the performance efficiency depends heavily on data distribution.

There have been important research results obtained based on APRIORI [6, 9]. Generally we can avoid vast range of search but not enough to sequence data. In general APRIORI based approach, when q is a *sub-pattern* of a pattern p , any sequence data that matches p should also match q . This property (called *anti-monotonicity* of patterns) provides us with the reduction of search space.

However, in a case of text mining, this is not true and we can't apply APRIORI technique any more. For example, in a text "*aabbba*", a is a sub-pattern of ab but we see a pattern ab matches 6 times, both a and b 3 times respectively and $[ab]$ 9 times. In other words, we can't have a naive counting method any more.

In this investigation, given an integer m and single long sequence data S , we discuss how to extract p where a new count $\mathcal{M}_S(p)$ is more than m . Here the *true problem* is the counting scheme \mathcal{M} . We extend some approach proposed[16] and discuss a new framework for disjunctive patterns within APRIORI.

In section 2 we formalize our problem, and, in section 3, we introduce a new measure for counting satisfying anti-monotonicity. Section 4 contains some experimental results.

² The national flag of Japan consists of red and white parts.

2 Text Mining from Single Sequence Data

In this work, we consider a word as a unit, called *item*. Any element in an itemset $I = \{i_1, \dots, i_L\}, L > 0$ is called *alphabet*, and a *sequence data* (or, text) $S = s_1 \dots s_m, m > 0$ is an ordered list of items, m is called a *length* of S , and S is called *m*-sequence. Note an item may appear many times in S .

A *disjunctive pattern* (or just a pattern) p is a form of $t_1 \dots t_n, n > 0$ where each t_i is an alphabet a or a disjunction $[a_1 a_2 \dots a_m], m > 0$, each a_j is a distinct alphabet.

Given two patterns $p = t_1 \dots t_n, n > 0$ and $q = v_1 v_2 \dots v_m, m \leq n$, we say q is a *sub-pattern* of p , denoted by $q \sqsubseteq p$, if there exist $1 \leq j_1 < \dots < j_m \leq n$ such that each v_k corresponds to t_{j_k} (denoted by $v_k \sqsubseteq t_{j_k}$ if no confusion arises) satisfying:

- If v_k is an alphabet a , $t_{j_k} = a$ or t_{j_k} is a disjunction containing a
- If v_k is a disjunction $[a_1 a_2 \dots a_m]$, we have both $t_{j_k} = [b_1 b_2 \dots b_l]$ and $\{a_1, \dots, a_m\} \subseteq \{b_1, \dots, b_l\}$

Example 1. ac is a sub-pattern of $abcd$. Similarly $[ac]$ is a sub-pattern of $[abcd]$, bd is a sub-pattern of $[ab]b[cd]de$, b is a sub-pattern of $[ab]$, and “ ac ” is a sub-pattern of $[ab][cd]$.

However, ab is not a sub-pattern of $[ab]$, nor $[ab]$ is a sub-pattern of ab .

We say a pattern $p = t_1 t_2 \dots t_n$ matches a sequence $S = c_1 c_2 \dots c_m$

- if t_1 is an alphabet a_1 , there exist $t_1 = a_1 = c_{i_1}, 1 \leq i_1 \leq m$ and the sub-pattern $t_2 \dots t_n$ matches $c_{i_1+1} \dots c_m$,
- and if t_1 is a disjunction $[a_1 a_2 \dots a_m]$, there exists a permutation $a_{j_1} \dots a_{j_m}$ of a_1, \dots, a_m that matches $c_1 \dots c_{i_1}$, and the subpattern $t_2 \dots t_n$ matches $c_{i_1+1} \dots c_m$.

Example 2. Assume S is $aabbba$. A pattern a matches S 3 times, ab 6 times and $[ab]$ 9 times. Note we can see more frequency by $[ab]$.

Given a sequence S , a function \mathcal{M}_S from patterns to non-negative integers satisfies *Anti Monotonicity* if for any patterns p, q such that $q \sqsubseteq p$, we have $\mathcal{M}_S(q) \geq \mathcal{M}_S(p)$. In the following, we assume some S and we say \mathcal{M} for \mathcal{M}_S .

Given \mathcal{M} and an integer $m > 0$ (called *minimum support*), A pattern p is called *frequent* if $\mathcal{M}(p) \geq m$. If \mathcal{M} satisfies anti-monotonicity, for any q such that $\mathcal{M}(q) < m$, there is no frequent p such that $q \sqsubseteq p$. By using this property, we can reduce search space to extract frequent patterns. In fact, this is the motivation of APRIORI[1] and the algorithm is given as follows:

- (1) Find frequent patterns of the minimum size
- (2) Find one-size larger patterns p where all the sub patterns are frequent. (The results are called *candidate*.) Stop if there is no pattern obtained.
- (3) Select all the frequent patterns from the candidates by scanning S and goto (2).

In this work, given a sequence S , we examine all the frequent patterns p . However, it is not easy to obtain \mathcal{M} satisfying anti-monotonicity. For example, “the number of matching” is not suitable as \mathcal{M} as shown in Example 2.

3 Anti-monotonic Counting

In this section, we discuss new counting methods for anti-monotonicity since naive matching is not suitable.

The first idea is called *head frequency*. Given a sequence $S = s_1s_2\dots s_r$ and a pattern p of $t_1t_2\dots t_n$, we define a *head frequency* $H(S, p)$ as follows:

$$H(S, p) = \sum_{i=1}^r Val(S, i, p)$$

where $Val(S, i, p)$ is 1 if the followings hold, and 0 otherwise:

Let $S(i)$ be a suffix of S from i -th position, i.e., $S(i) = s_i\dots s_r$. If t_1 is an alphabet a , we have $s_i = a$ and $t_2t_3\dots t_n$ matches $S(i + 1)$. And if t_1 is a disjunction $[a_1a_2\dots a_m]$, there exists j such that $s_i = a_j$ (for instance, $j = 1$), and $[a_2a_3\dots a_m]t_2\dots t_n$ matches $S(i + 1)$.

Intuitively $H(S, p)$ describes the number of matching of p from the heading of S or its suffix.

Example 3. (1) Let S be a sequence **bbba**. If $p = ba$, we have $H(S, p) = 3$ while p matches S 3 times. If $p = a$, we have $H(S, p) = 1$ and the number of matching is 1. By the definition, we have $a \sqsubseteq ba$ but not $H(S, a) > H(S, ba)$.

(2) Let S be **aabbba**. If $p = ab$, we have $H(S, p) = 2$ and the number of matching is 6. If $p = ba$, we see $H(S, p) = 3$ and p matches S 3 times. If $p = [ab]$, we get $H(S, p) = 5$ and the pattern matches S 9 times. Finally if $p = a$, we have $H(S, p) = 3$ and the number of matching is 3. By the definition, $a \sqsubseteq [ab]$ holds but $H(S, a) > H(S, [ab])$ doesn't.

As shown in this example, the head frequency $H(S, p)$ doesn't satisfy anti-monotonicity. Note that this counting ignores matching appeared in the subsequent sequence. That's why we introduce a new counting $D(S, p)$, called *total frequency*, which means the minimum $H(S, q)$ for any $q \sqsubseteq p$.

$$D(S, p) = \text{MIN}\{H(S, q) | q \sqsubseteq p\}$$

Theorem 1. $D(S, p)$ satisfies anti-monotonicity.

Proof. Given patterns p, q such that $q \sqsubseteq p$, we must have $D(S, p) = \text{MIN}\{H(S, r) | r \sqsubseteq p\}$ and $D(S, q) = \text{MIN}\{H(S, r) | r \sqsubseteq q\}$ by the definition. That means $D(S, q) \geq D(S, p)$ since $q \sqsubseteq p$. (*Q.E.D.*)

Example 4. (1) Let S be **bbba**. If $p = ba$, we have $D(S, p) = 1$. And if $p = a$, we see $D(S, p) = 1$.

(2) Let S be **aabbba**. If $p = ab$ we have, $D(S, p) = 2$ Cif $p = ba$, we see $D(S, p) = 3$, if $p = [ab]$, we get $D(S, p) = 3$ Cand if $p = a$, we have $D(S, p) = 3$.

(3) Let S be *caabbbbc*. If $p = ab$, we have $H(S, p) = 2$ and $D(S, p) = 2$. If $p = ac$, we have $H(S, p) = 2$ and $D(S, p) = 2$. And if $p = [ac]$, we see $H(S, p) = 3$ and $D(S, p) = 2$ while p matches S 4 times. In these cases, sub-patterns (i.e., a, c) of ac and $[ac]$ appear more interspersed in S and this is why *total frequency* is different from *head frequency* and the number of matching.

According to theorem 1, to obtain total frequency of p of length n , it is enough to examine head frequency of all the sub-patterns. And, as the next theorem says, it is enough to calculate the ones for all the suffixes of p in ascending order of length. Note there are only n suffixes of p thus we can reduce search space dramatically.

Theorem 2. For a sequence S and a pattern p , $D(S, p) = \text{MIN}\{H(S, p), D(S, p(2))\}$.

Proof. Let S be a sequence $s_1s_2\dots s_m$, p a pattern $t_1t_2\dots t_n$ and $p(i)$ be a sub-pattern (suffix) $t_i\dots t_n$. We show $D(S, p) = \text{MIN}\{H(S, p(i)) | i = 1, \dots, n\}$.

Let $q = u_1u_2\dots u_k$ be any sub-pattern of p . By the assumption, there exist $1 \leq j_1 < \dots < j_k \leq n$ such that $u_i \sqsubseteq t_{j_i k}$, $i = 1, \dots, k$.

We define the expansion q' of q at a position i as follows ($i > 0$).

q' is one of the followings

- (i) $u_1u_2\dots u_i v u_{i+1} \dots u_k$, that is, v is inserted just after u_i .
- (ii) $u_1u_2\dots u'_i \dots u_k$, that is, u_i is replaced by u'_i such that $u_i \sqsubseteq u'_i$.

We show $H(S, q) \leq H(S, q')$, that is, we show that $Val(S, i, q') = 1$ means $Val(S, i, q) = 1$. If q' is an expansion of (i), we have $Val(S, i, q) = 1$ by ignoring v at matching step. And, in the case of (ii), we have $Val(S, i, q) = 1$ by ignoring $u'_i - u_i$ part at matching step.

Because t_{j_1} can be obtained by expansions of q , we must have $H(S, q) \geq H(S, t_{j_1})$. This means, for any sub-pattern q of p , there exists $t_{j'}$ that has smaller head frequency. (*Q.E.D.*)

4 Experimental Results

4.1 Experimental Data

As a test collection for our experiments, we discuss NTCIR-3 PATENT (IR Test Collection). The collection contains several data but we discuss only PAJ English Abstract (1995). Documents in JAPIO Japanese abstracts (1995-1999) are translated into PAJ English Abstract of 4GB in total. In this experiment, we select 1000 articles in 1995 abstracts that are kept in timestamp order. We remove all the stop-words and do stemming[4] to each articles.

Here are some examples of the information (after stemming):

```
control adjust speed thresh depth regul grain culm state oper load
combin shape initi puls power high load devic regul control thresh
depth grain culm detect thresh depth sensor
```

Table 1. Number of Frequent Patterns from Patent Data

n -pattern	our method	$T - freq$ method
$n = 1$	207(0)	207
$n = 2$	121(76)	45
$n = 3$	3(1)	2
$n = 4$	0(0)	0

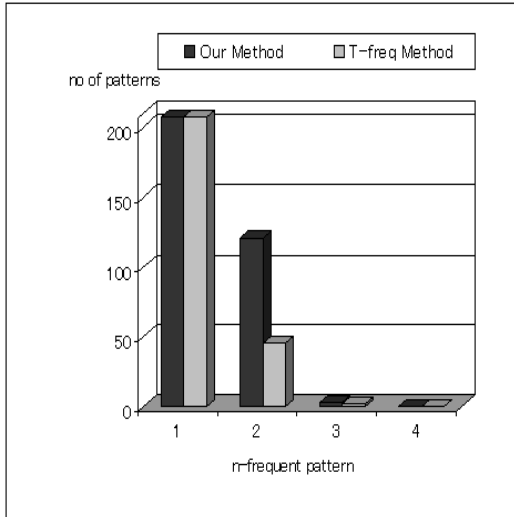


Fig. 1. Number of Frequent Patterns from Patent Data

4.2 Preliminaries

In this investigation we extract frequent patterns base on APRIORI algorithm described previously[1]. To evaluate our results, we compare $T - freq(S,p)$ method [16] with us. The minimum support is 20 which is 2%. In this experiment we manipulate disjunctive patterns of $[ab]$, $[ab]c$, $[abc]d$, ..., that is, we discuss only patterns where one disjunction appears, and we ignore patterns with multiple disjunctions such as $[ab]c[de]$.

4.3 Results

Here is the result shown in Table 1 and Figure 1. In the table, we show the numbers of frequent patterns extracted by the two methods and the distribution in the figure. Note that the number in “(...)” means the number of extracted patterns by our method but not by $T - freq$ method.

Here are some examples extracted by our method:

```
([medicin,patient]),([prepar,medicin]),([surfac, sheet])...
([prepar,medicin]patient)...
```

And the next are obtained by $T - freq$ method:

```
(medicin,prepar),(medicin,patient),(devic,capabl)...
(provid,capabl,devic)...
```

4.4 Discussion

By examing the table 1, we can extract more patterns in $n = 2$ and $n = 3$.

More important is that we can obtain 76 disjunctive patterns ($n = 2$) only by our method. This is because disjunctive patterns match S more times thus we can see more frequency. For example, we have $[surfac, sheet]$ with the frequency 22 and $[prepar, medicin]patient$ with the frequency 21, which are ignored by $T - freq$ method.

5 Related Works

There have been several approach to obtain sequence patterns[2, 15], mainly based on APRIORI algorithm[1]. Their issue is to extract sequence patterns that appear in many sequences, given a set of sequences that consist of lists of itemsets. The problem is combinatorial but efficient algorithms have been proposed by introducing specialied data structures such as FreeSpan [5] and PrefixSpan [13], by means of lattice theory such as SPADE [17]. Some frameworks for *regular expression* have been also proposed such as SPIRIT [3] and a kind of Constraint Satisfaction Problem[7]. But in these approach the expression plays as a constraint and not targeted patterns like us. More important is that all of them discuss *multiple* sequences and that the frequency is defined in terms of the number of sequences containing the pattern. We see this is not really useful for text mining.

The problem of frequent patterns on single sequence is found in *episode* [8] where sequence is defined as a list of events. The problem for text mining is found in [16] but they don't discuss disjunctive patterns.

6 Conclusion

In this investigation, we have proposed disjunctive patterns and total frequency for the purpose of efficient pattern mining on single sequence data. The counting method satisfies anti-monotonicity thus we can obtain efficient mining algorithm based on APRIORI. By some experiments we have shown the effectiveness of the proposed method compared to traditional approach, especially we have shown more frequent patterns extracted. Now we extend our approach to more powerful expression to get potential patterns.

Acknowledgment

We would like to acknowledge the financial support by Grant-in-Aid for Scientific Research (C)(2) (No.16500070) from Ministry of Education, Culture, Sports, Sci-

ence and Technology in Japan. Also we would like to acknowledge the support of NTCIR-3 PATENT (IR Test Collection) from National Institute of Informatics (NII) in Japan.

References

1. Agrawal, R. and Srikant, R.: Fast Algorithm for Mining Association Rules, proc. VLDB, 1994, pp.487-499
2. Agrawal, R. and Srikant, R.: Mining Sequential Patterns, proc. ICDE, 1995, pp.3-14
3. Garofalakis, M., Rastogi, R. and Shim, K.: SPIRIT : Sequential Pattern Mining with Regular Expression Constraints, proc. VLDB, 1999, pp.223-234
4. Grossman, D. and Frieder, O.: Information Retrieval – Algorithms and Heuristics, Kluwer Academic Press, 1998
5. Han, J., Pei, J., Mortazavi, B., Chen, Q., Dayal, U. and Hsu, M-C.: FreeSpan: Frequent Pattern-Projected Sequential Patterns Mining, proc. KDD, 2000, pp.355-359
6. Han, J. and Kamber, M.: Data Mining : Concepts and Techniques, Morgan Kaufmann, 2000
7. Albert-Lorincz, H. and Boulicaut, J-F.: Mining Frequent Sequential Patterns under Regular Expressions: A Highly Adaptive Strategy for Pushing Constraints, proc. SIAM DM, 2003, pp.316-320
8. Mannila, H. and Toivonen, H. and Verkamo, I.: Discovery of Frequent Episodes in Event Sequences, *Data Mining and Knowledge Discovery* 1(3), 1997, pp.259-289
9. Hand, D., Mannila, H. and Smyth, P.: Principles of Data Mining, MIT Press, 2001
10. Motoyoshi, M., Miura, T., Watanabe, K. and Shioya, I.: Temporal Class Mining for Time Series Data, proc. CIKM, 2002
11. Motoyoshi, M., Miura, T. and Shioya, I.: Clustering by Regression Analysis, International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2003), pp.202-211 (Springer LNCS 2737), 2003, Prague
12. Nagao, M.: Natural Language Processing (in Japanese), Iwanami, 1996
13. Pei, J., Han, J., Mortazavi, B., Pinto H., Chen, Q., Dayal, U. and Hsu, M-C.: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth, proc. ICDE, 2001, pp.215-224
14. Sebastiani, F.: Machine learning in automated text categorization, ACM Comp. Survey 34-1, 2002, pp.1-47
15. Srikant, R and Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements, proc. EDBT, 1996, pp.412-421
16. Takano, Y., Iwanuma, K. and Nabeshima, H.: A Frequency Measure of Sequential Patterns on a Single Very-Large Data Sequence and its Anti-Monotonicity, in Japanese, proc. FIT, 2004, pp.115-118
17. Zaki, M.J.: Efficient Enumeration of Frequent Sequences, proc. CIKM, 1998, pp.68-75

Mining Expressive Temporal Associations from Complex Data

Keith A. Pray¹ and Carolina Ruiz²

¹ BAE Systems, Burlington, MA 01803 USA
keith.pray@baesystems.com

² Department of Computer Science, Worcester Polytechnic Institute (WPI),
Worcester, MA 01609 USA
ruiz@cs.wpi.edu
<http://www.cs.wpi.edu/~ruiz>

Abstract. We introduce an algorithm for mining expressive temporal relationships from complex data. Our algorithm, AprioriSetsAndSequences (ASAS), extends the Apriori algorithm to data sets in which a single data instance may consist of a combination of attribute values that are nominal sequences, time series, sets, and traditional relational values. Data sets of this type occur naturally in many domains including health care, financial analysis, complex system diagnostics, and domains in which multi-sensors are used. AprioriSetsAndSequences identifies predefined events of interest in the sequential data attributes. It then mines for association rules that make explicit all frequent temporal relationships among the occurrences of those events and relationships of those events and other data attributes. Our algorithm inherently handles different levels of time granularity in the same data set. We have implemented AprioriSetsAndSequences within the Weka environment [1] and have applied it to computer performance, stock market, and clinical sleep disorder data. We show that AprioriSetsAndSequences produces rules that express significant temporal relationships that describe patterns of behavior observed in the data set.

1 Introduction

This paper extends the use of association rules [2] to complex temporal relationships essential to many domains. Our association rule mining approach discovers patterns in data sets whose instances consist of any combination of standard, set-valued, and sequential attributes. These data sets occur naturally in several scientific, engineering, and business domains, and are generally richer than the transactional, the relational, and the sequence data sets to which association rule mining has been applied. To date, our mining approach has been applied to computer system performance, stock market analysis and clinical sleep disorder data [3]. Complex system diagnostics, network intrusion detection, and medical monitoring are some related domains to which this work can also be applied.

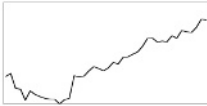
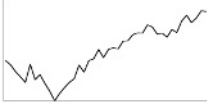
ID	CPU %	CPU (MHz)	memory %	algorithms
1		600		{neural net, back-propagation}
2	40, 52, 67, 80, ...	600	10, 26, 46, 69, 86, ...	{C4.5}
3	10, 39, 87, 96, ...	300	19, 50, 82, 80, 70 ...	{naive Bayes}
...

Fig. 1. A sample of a data set containing complex instances. Here, sequential values are represented in a graphical manner only for the first row to save space

A main motivation for mining these temporal rules from complex data comes from our previous experience working on the performance analysis of a hardware and software backup/restore product. The time it takes to complete a backup can be of great importance. One of the most labor consuming tasks is to predict this time. A rule based expert system was built as a way to disseminate the performance knowledge used to predict backup time. This expert system quickly grew to include over six hundred rules. The need for automating the discovery of new rules was apparent.

Figure 1 depicts a small sample of the type of complex data set that our mining approach applies to. In this computer performance data set, each instance (row) corresponds to a single test in which a process was run until it completed a task. The attributes describe the conditions under which the test was run and state information collected during the test including standard attributes such as processor speed and physical memory size; set-valued attributes such as which algorithms the process ran; and sequential attributes such as the CPU utilization percentage, memory usage, and the total number of processes running over time. Other such numeric and nominal sequential attributes include CPU and memory usage of all other processes, I/O activity on main storage devices, memory paging, and process state (running, exit normally, exit without output). All time sequence attributes in a single data instance share the same time line.

Events of interest in the sequential attributes in this data set include among others *increases* and *decreases* in the utilization of a resource (CPU, memory), going above/below a certain usage threshold, and whether or not a process terminates normally. A sample interesting temporal pattern in this domain is: “Java processes running a machine learning algorithm that are not given the -P option, that exhibit an increase in its memory usage, and that during this increase its memory paging also increases tend to, with likelihood 82%, exit prematurely”. This temporal pattern is captured by our association rule:

$$\text{process(java)-mem-usage-increase } [t_0, t_2] \ \& \ \text{process(java)-page-increase } [t_1, t_2] \ \& \ \text{flag=-P-missing} \Rightarrow \text{process(java)-exit-without-output } [t_3, t_4], \text{ conf}=82\%.$$

here, $t_0, t_1, t_2, t_3,$ and t_4 are *relative* time indices that are used to express the relative order in which these events are observed.

The algorithm presented here to mine these temporal relationships, *AprioriSetsAndSequences (ASAS)*, takes as input a complex temporal data set as described, a set of predefined event types, and the standard Apriori minimum support and minimum confidence parameters. An *event type* is a template of a subsequence of interest in a time sequence attribute. Our mining algorithm starts by identifying occurrences of these event types in the corresponding sequence attributes as described in Section 2. Then, it generates all the frequent relationships among the temporal occurrences of these events and among these and the other non-temporal attributes. Section 3 describes the algorithm. Since there are 13 ways of sorting just two events in time (*before, after, overlaps,* etc.) [4], and the number of possible orderings of a set of events grows exponentially with the number of events in the set, central issues in the design and implementation of our mining algorithm are the strategy used to prune unnecessary orderings from consideration, and the data structures used to effectively handle potentially frequent orderings of events. We describe these in Section 3. That section also describes our extensions of the basic notions of support and confidence needed to handle multiple occurrences of an event or a pattern in a complex data instance. Section 4 presents an evaluation of our mining algorithm in the stock market domain. Section 5 surveys related work and contrasts our approach to others. Section 6 summarizes the contributions of this paper and discusses future work.

2 Identifying Events in Sequences

Events of interests are available in multiple domains. Examples of those are *head & shoulders reversal* and *ascending triangle* in stock market analysis [5], and *increase in CPU utilization* in the performance domain. An event can be described by a Boolean condition or by a template of the event “shape”. Such a template can be for example a 2-dimensional curve.

Given a collection of domain-specific events of interest, we identify occurrences of those events in the sequential attributes. Once the occurrences of an event of interest have been identified in the values of a sequential attribute, they are stored in a new event set attribute. We define an *event set attribute* as an attribute whose values are sets of an event type occurrences. As an example, consider an event attribute for the CPU time sequence attribute. The CPU time sequence attribute is the percentage of CPU usage in the overall computer system. This event attribute specifies when the CPU usage increases. Assume that in a particular data set instance I , increases in CPU usage occur from time 2 to 8, 13 to 19, and 35 to 47. Then, the I value for the new attribute CPU-Increase is $\{ [2,8], [13,19], [35,47] \}$. *AprioriSetsAndSequences* mines temporal associations directly from events. This keeps intact the temporal information represented in the data set while eliminating much of the work involved in scanning the actual sequences during mining. The events are akin to indexes into the sequences.

3 The ASAS Algorithm

Input. ASAS takes as input a data set consisting of instances, and minimum support and minimum confidence thresholds. Each instance has a set of attributes. Attributes can be of type nominal, set, and event set. An event set is simply a set whose elements are events.

Handling Set Attributes. We use the algorithm to mine association rules from set valued data described in [6], which was implemented within Weka [7].

Handling Event Attributes. Since we are not interested in the occurrence of say a CPU-Increase event at absolute time [13, 18], but rather on the fact that this event occurred in a relative temporal position with respect to other events, ASAS uses a relative time line t_0, t_1, t_2, \dots . There are no units implied. Each point on the relative time line corresponds to the begin times or end times of one or more events in the item set. When an event is added to an item set the item set's relative time line must be updated. As an illustration, the real time instance {Disk Increase [5,25], CPU Increase [10,40]} is represented by the relative time item set {Disk Increase $[t_0, t_2]$, CPU Increase $[t_1, t_3]$ }. Adding the event Memory Sustain with real times [2,35] to the item set results in {Disk Increase $[t_1, t_3]$, CPU Increase $[t_2, t_5]$, Memory Sustain $[t_0, t_4]$ }. Simply sorting the real time values and numbering them starting from t_0 yields the relative times.

Level 1 Candidate Generation. Our ASAS algorithm creates an item set of size one for each of the *regular* (i.e. non-event) items appearing in the data set, as Apriori does. For event items, it generates representative items. For instance, the representative of Memory Sustain [2,35] is Memory Sustain $[t_0, t_1]$, which represents all the Memory Sustain events regardless of their real time occurrences.

Level 1 Counting Support. The support of an item set is the percentage of instances in the data set that contain the item set. The weight of an item set is the number of instances that contain it. For a regular item, we count its support as the Apriori algorithm does. For an event item, say Memory Sustain $[t_0, t_1]$, a data instance contributes to its weight (and hence to its support) if the instance contains an occurrence of the item, say Memory Sustain [2,35].

Level 2 Candidate Generation. As Apriori, ASAS generates candidate item sets of size two by combining each pair of frequent item sets of size one. However, for each pair of event items there exist thirteen different item sets that represent the possible temporal relationships [4] between the two event items.

Level 2 (and up) Counting Support. A data instance contributes to the weight of an item set if it contains all the regular and the event items in the item set. If the item set contains only one event item, checking if the instance contains it is done as described above in Level 1 Counting Support. If the item set contains more than one event item then a mapping must exist from event items in the item set to event items in the instance. This mapping provides a match (or unification) from the relative times in the item set to the real times in the data instance that preserves the relative order of occurrence of the events. For example, a data

instance that contains {Disk Increase [5,25], Disk Increase [40,55], CPU Increase [50,60]} counts towards the weight (and hence the support) of the item set {Disk Increase $[t_0, t_2]$, CPU Increase $[t_1, t_3]$ } with mapping $t_0 \rightarrow 40$, $t_1 \rightarrow 50$, $t_2 \rightarrow 55$, $t_3 \rightarrow 60$. Our ASAS algorithm uses a specially designed data structure to make the search for a valid such mapping very efficient.

Level 3 (and up) Candidate Generation. During the combination of item sets, regular items are handled as Apriori does. Two item sets containing event items can be combined if there exists a mapping from all the event items in the first item set to all the event items in the second item set. This excludes event items that are the last item in an item set (items in an item set are sorted using an arbitrary but fixed order). As for level two candidate generation, it is possible for more than one candidate item set to be generated for each pair of frequent item sets that are combined. The algorithm for generating these possibilities is more involved than the iteration of thirteen possible temporal relationships. Consider combining the item sets A and B :

$$\begin{aligned} A: & \{ \text{Disk Increase } [t_0, t_2], \text{CPU Increase } [t_1, t_3] \} \\ B: & \{ \text{Disk Increase } [t_1, t_2], \text{Memory Sustain } [t_0, t_3] \} \end{aligned}$$

These item sets can be combined since they have the same number of items, a mapping exists between the Disk Increase event in item set A and the Disk Increase event in item set B , and the event items listed last in A and B are different. The temporal relationship between the CPU Increase event in A and the Memory Sustain event in B is not known. Some of the possible relationships can be eliminated by inferring information from the fact that the Disk Increase event in both A and B is the same event. Combining the two item sets is done by adding the last item from the first item set to the second item set. All possible begin and end time pairs of the Memory Sustain event need to be determined in relation to item set A 's existing relative time line. A candidate item set is generated for each pair. In these candidate item sets the relative time line is renumbered starting from t_0 .

$$\begin{aligned} & \{ \text{Disk Increase } [t_1, t_3], \text{CPU Increase } [t_2, t_5], \text{Memory Sustain } [t_0, t_4] \} \\ & \{ \text{Disk Increase } [t_1, t_3], \text{CPU Increase } [t_2, t_4], \text{Memory Sustain } [t_0, t_4] \} \\ & \{ \text{Disk Increase } [t_1, t_3], \text{CPU Increase } [t_2, t_4], \text{Memory Sustain } [t_0, t_5] \} \end{aligned}$$

Algorithm.

- 1: given a data set of instances DS , and minimum weight $minW$
- 2: **for all** regular items i in DS **do**
- 3: create candidate item set c of size $k = 1$
- 4: add i to c and add c to candidate list C
- 5: **for all** event items e in data set **do**
- 6: **if** event type of e not present in event type list ET **then**
- 7: create candidate item set c of size $k = 1$
- 8: create a new event item ei with event type of e and begin time = 0 and end time = 1
- 9: add ei to c , add c to C , and add e to ET

```

10: for all instances  $I$  in  $DS$  do
11:   for all  $c$  in  $C$  do
12:     if  $I$  contains the item in  $c$  then
13:       increment weight of  $c$ 
14:   for all  $c$  in  $C$  do
15:     if weight of  $c \geq minW$  then
16:       add  $c$  to frequent item sets of size  $k$  list
17:   remove all from  $C$ 
18:   while frequent item set of size  $k$  list is not empty do
19:      $k++$ 
20:   for all pairs of item sets  $f1$  and  $f2$  in the frequent item sets of size  $k-1$  list do
21:     if  $f1$  and  $f2$  contain event items then
22:       generate 13 candidates, 1 for each possible temporal relationship between
       the event items  $f1$  and  $f2$  do not have in common
23:     else
24:       generate 1 candidate by combining  $f1$  and  $f2$ 
25:       add generated candidate(s) to  $C$ 
26:   for all instances  $I$  in  $DS$  do
27:     for all  $c$  in  $C$  do
28:       if all regular items  $i$  in  $c$  are included in  $I$  then
29:         if mapping exists between all event items  $ei$  in  $c$  to event items in  $I$  such
         that all temporal relationships are the same then
30:           increment weight of  $c$ 
31:   for all  $c$  in  $C$  do
32:     if weight of  $c \geq minW$  then
33:       add  $c$  to frequent item sets of size  $k$  list
34:   remove all from  $C$ 

```

Rule Construction and Confidence Calculation. The construction of rules from frequent item sets is similar to that of the Apriori algorithm, with the exception of the confidence calculation. Traditionally confidence is defined for a rule $A \Rightarrow B$ as the percentage of instances that contain A that also contain B . That is, $\text{support}(AB)/\text{support}(A)$. Consider a data set that has one time sequence: $\langle a, b, a, a, a, a \rangle$. Since there is one instance in our data set and it contains the item set $\{a[t_0, t_1], b[t_2, t_3]\}$, the support of each of the item sets $\{a[t_0, t_1]\}$, $\{b[t_2, t_3]\}$, and $\{a[t_0, t_1], b[t_2, t_3]\}$ is 100%. If support were used to calculate the confidence of the rule $a[t_0, t_1] \Rightarrow b[t_2, t_3]$, it would be 100%. This implies that 100% of the time a appears, b follows. Looking at the time sequence, only 20% of the time is a followed by b . We define *the confidence of a rule containing event items as the percentage of mappings from the antecedent of the rule to the data instances that can be extended to mappings from the full set of items in the rule to the same data instances*. Note that there are 5 possible mappings from the antecedent $a[t_0, t_1]$ of the rule to the data instance, but only one of them can be extended to a mapping from $\{a[t_0, t_1], b[t_2, t_3]\}$ to the instance. Hence, the confidence of this rule is $1/5$ or 20%.

4 Empirical Evaluation

We have applied our ASAS algorithm to different domains including computer system performance, stock market analysis, and clinical sleep disorder data. Due to space limitations we include here only some results on Stock Market data analysis. The data used [8] consists of ten years worth of closing prices from 7 technology companies from 1992 to 2002 obtained from Yahoo! Finance. Additionally, events such as new product releases, awards received, negative press releases, and expansions or mergers from each company were obtained from each respective company's web site. Each of the 24 instances in this data set represents a single quarter year. All 10 years are not represented because information on the additional events listed above were not available for all years. Before mining, the sequences of closing prices for a quarter for each company are filtered for events. The financial events detected include rounded top, selling climax, ascending triangle, broadening top, descending triangle, double bottom, double top, head & shoulders, inverse head & shoulders, panic reversal, rounded bottom, triple bottom, triple top, sustain, increase, and decrease [5].

Rules. Numerous interesting rules were found by our ASAS algorithm. Due to space limitations we show here just the pair of rules below. They have the same events in them but one has a predictive form (i.e., the events in the consequent occur later in time than the events in the antecedent) and the other has a diagnostic form (i.e., the events in the consequent occur before the events in the antecedent).

CSCO Expand Merge $[t_4, t_5]$ & AMD Ascending Triangle $[t_0, t_1]$
 \Rightarrow SUNW Sustain $[t_2, t_3]$ [Conf: 0.91, Sup: 0.42, Event Weight: 10]

AMD Ascending Triangle $[t_0, t_1]$ & SUNW Sustain $[t_2, t_3]$
 \Rightarrow CSCO Expand Merge $[t_4, t_5]$ [Conf: 1.0, Sup: 0.42, Event Weight: 11]

CSCO Expand Merge 1-7 days, AMD Ascending Triangle 6-30 days, SUNW Sustain 6-13 days

Advanced Micro Devices Inc's closing stock prices exhibits an ascending triangle pattern for 6 to 30 days. Sometime after but during the same quarter Sun Microsystems Inc's closing stock price remains fairly constant for 6 to 13 days. Sometime after in the same quarter Cisco goes through a period of expansion or merger for 1 to 7 days. The predictive form of the rule has a 100% confidence. In any quarter in the data set, every time AMD and Sun exhibit the behaviors described, Cisco expands or merges.

ASAS Performance. Figure 2 shows the seconds used to mine rules per frequent item set found and other metrics for slightly differing data sets from the stock market domain. The total time it takes to mine appears to be insensitive to the number of event attributes, the number of event occurrences, and the average length of the time line. It seems only the number of frequent item sets found in a data set greatly increases mining time. The time spent finding each frequent item set seems related to the number of event occurrences and the number of event attributes in the data set.

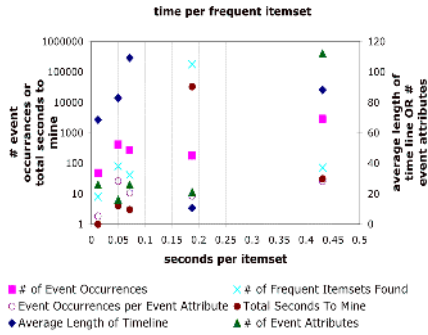


Fig. 2. Various Metrics

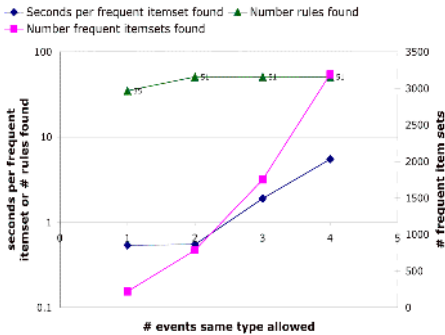


Fig. 3. 49 Percent Support

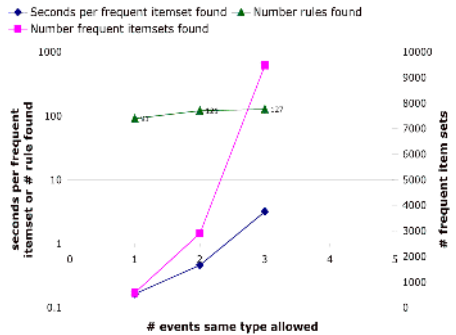


Fig. 4. 40 Percent Support

Figure 3 shows the results of varying the maximum number of events with the same type that can appear in a rule. This was done with a support setting of 49%. 16 rules containing 2 events of the same type were found. Beyond a maximum of 2 more time is spent per frequent item set with no additional rules found to justify the cost. The lower the percentage of new rules found by increasing the maximum number of events of the same type allowed, the more time per frequent item set will be spent during mining. Figure 4 shows results using a support of 40%. Although more rules are found due to the lower support, more time is spent per frequent item set.

5 Related Work

There has been a great deal of interest in devising approaches to mine associations from sequential data. These approaches can be roughly divided into two groups. The first group contains approaches that extend the Apriori algorithm to sequences. These approaches assume data instances that are sequences of commercial transactions. A commercial transaction is called an *event*. These ap-

proaches mine frequent patterns from those data instances. Among others, the work by Srikant and Agrawal [9] and by Zaki [10] and collaborators belong to this group. They use the notions of *time window* and *max/min gaps* to address the complexity of the mining task. Zaki [10] considers item set constraints for this same purpose. One difference between our work and the approaches in this group is that our notion of *event* is a non-trivial time interval and theirs is a point in time (instantaneous events). This has a profound impact on the expressiveness of our association rules and on the complexity of the mining process, as in our case the possible orderings of two single events is 13 while for them that number of orderings is only 3. Another important difference is that in our approach we consider data instances that are combinations of several attribute types, while their instances are sequences of transactions.

The second group of association rule approaches to sequential mining includes the work by Mannila et al. [11, 12, 13]. They consider *episodes* of events, where events are again points in time. Episodes are collections of partially ordered events that occur close to each other in time. This constraint addresses the complexity of the search in a way similar to the time window approach described above. Our work extends theirs by allowing events that are time intervals. This enhances the collection of partial orders that are applicable to a set of events and thus the expressiveness of the mined patterns.

Roddick and Spiliopoulou [14] provide an excellent survey of temporal knowledge discovery. Rainsford and Roddick [15] report efforts on extending association rules with temporal information. Their work is similar to ours in that they also consider the 13 possible ways in which two temporal events can be ordered in time. However, the expressiveness of their association rules is very restricted in comparison with ours. Bettini et al. [16] describe an approach to mine temporal associations that allows the user to define a rule template, and their algorithm finds valid instantiations of the rule template in the data set. Our approach is more general than theirs in that the user is not restricted to use just one temporal template for each mining task, as our algorithm considers all possible temporal patterns that are frequent. Also, *we can explore several time-granularities during the same mining task, just by defining an event-based attribute for each relevant time-granularity and letting them “intersect” with other events of interest.* Other approaches that employ user-defined temporal templates are those described by Han and collaborators [17, 18]. Their multidimensional intertransaction association rules are particular cases of our complex temporal association rules.

6 Conclusions and Future Work

We introduce an algorithm for mining expressive temporal relationships from complex data sets in which a single data instance may consist of a combination of attribute values that are nominal sequences, time series, sets, and traditional relational values. Our mining algorithm is close in spirit to the two-stage Apriori algorithm. Our work contributes to the investigation of prune strategies and

efficient data structures to effectively handle the added data complexity and the added expressiveness of the temporal patterns. Furthermore, the work described here provides a foundation for future investigation and comparison of alternate measures of item set interestingness and alternate frequent item sets search techniques as those discussed in the association rule mining literature but in the context of complex data.

References

1. Frank, E., Witten, I.H.: *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers (2000)
2. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: *Proc. of the ACM SIGMOD Conf. on Management of Data*, Washington, D.C., ACM (1993) 207–216
3. Laxminarayan, P., Ruiz, C., Alvarez, S., Moonis, M.: Mining associations over human sleep time series. In: *Proc. 18th IEEE Intl. Symposium on Computer-Based Medical Systems*, Dublin, Ireland, IEEE (2005)
4. Allen, J.: Maintaining knowledge about temporal intervals. *Communications of the ACM* **26** (1983)
5. Little, J., Rhodes, L.: *Understanding Wall Street*. 3rd edn. Liberty Hall Press and McGraw-Hill Trade (1991)
6. Shoemaker, C., Ruiz, C.: Association rule mining algorithms for set-valued data. In Liu, J., Cheung, Y., Yin, H., eds.: *Proc. of the Fourth Intl. Conf. on Intelligent Data Engineering and Automated Learning*, Lecture Notes in Computer Science. Vol. 2690. Springer-Verlag (2003) 669–676
7. Stoecker-Sylvia, Z.: Merging the association rule mining modules of the Weka and ARMiner data mining systems. Undergraduate Thesis. WPI (2002)
8. Holmes, S., Leung, C.: Exploring temporal associations in the stock market. Undergraduate Thesis. WPI (2003)
9. Agrawal, R., Srikant, R.: Mining sequential patterns. In: *Intl. Conf. on Database Engineering*, IEEE (1995) 3–14
10. Zaki, M.: Sequence mining in categorical domains: Incorporating constraints. In: *Proc. Intl. Conf. on Information and Knowl. Management (CIKM)*. (2000) 422–429
11. Mannila, H., Toivonen, H., Verkamo, A.I.: Discovering Frequent Episodes in Sequences. In Fayyad, U.M., Uthurusamy, R., eds.: *Proc. of the First Intl. Conf. on Knowledge Discovery and Data Mining (KDD-95)*, Montreal, Canada (1995)
12. Mannila, H., Toivonen, H.: Discovering generalized episodes using minimal occurrences. In: *Proc. of the Second Intl. Conf. on Knowledge Discovery and Data Mining (KDD'96)*, Portland, Oregon, AAAI Press (1996) 146–151
13. Das, G., Lin, K.L., Mannila, H., Renganathan, G., Smyth, P.: Rule discovery from time series. In: *Proc. of the 4th Intl. Conf. on Knowledge Discovery and Data Mining*, ACM (1998) 16–22
14. Roddick, J., Spiliopoulou, M.: A survey of temporal knowledge discovery paradigms and methods. *IEEE Trans. on Knowledge and Data Engineering* **14** (2002) 750–767
15. Rainsford, C., Roddick, J.: Adding temporal semantics to association rules. In: *Proc. of the Third European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'99)*. (1999) 504–509
16. Bettini, C., Sean Wang, X., Jajodia, S.: Testing complex temporal relationships involving multiple granularities and its application to data mining. In: *Proc. of the Fifteenth ACM Symposium on Principles of Database Systems*. (1996) 68–78

17. Tung, A.K., Lu, H., Han, J., Feng, L.: Efficient mining of intertransaction association rules. *IEEE Trans. on Knowledge and Data Engineering* (2003) 43–56
18. Lu, H., Feng, L., Han, J.: Beyond intratransaction association analysis: mining multidimensional intertransaction association rules. *ACM Trans. on Information Systems (TOIS)* **18** (2000) 423–454

Statistical Supports for Frequent Itemsets on Data Streams

Pierre-Alain Laur¹, Jean-Emile Symphor¹, Richard Nock¹,
and Pascal Poncelet²

¹ GRIMAAG-Dépt Scientifique Interfacultaire, Université Antilles-Guyane,
Campus de Schoelcher, B.P. 7209, 97275 Schoelcher Cedex,
Martinique, France

{palaaur, je.symphor, rnock}@martinique.univ-ag.fr

² LG2IP-Ecole des Mines d'Alès, Site EERIE,
parc scientifique Georges Besse, 30035 Nîmes Cedex, France
pascal.poncelet@ema.fr

Abstract. When we mine information for knowledge on a whole data streams it's necessary to cope with uncertainty as only a part of the stream is available. We introduce a stastistical technique, independant from the used algorithm, for estimating the frequent itemset on a stream. This statistical support allows to maximize either the precision or the recall as choosen by the user, while it doesn't damage the other. Experiments with various association rules databases demonstrate the potential of such technique.

1 Introduction

A growing body of works arising from researchers in Databases and Data Mining deals with data arriving in the form of continuous potentially infinite streams. Many emerging and real applications generate data streams: trend analysis, fraud detection, intrusion detection, click stream, among others. In fraud detection, data miners try to detect suspicious changes in user behavior [5]. Trend analysis is an important problem that commercial applications have to deal with [8]. Security of network systems is becoming increasingly important as more and more sensitive informations are being stored. Intrusion detection has thus become a critical approach to protect systems [7].

From now on, we consider *items* to be the unit information, and *itemsets* to be sets of items. An itemset is θ -frequent if it occurs in at least a fraction θ of the stream (called its support), where θ is a user-specified parameter. As the item flow is fast and represent a huge amount of information, it prevents its exact storage. Out of the uncertainty it generates, the problem becomes to store the information so as to keep valid its most crucial contents. One example of such a content is the list of the most frequent items of itemsets encountered, a crucial issue in Data Mining that has recently attracted significant attention [6, 10, 12, 7].

When the database is subject to be updated regularly, maintaining frequent itemsets has been successfully addressed by various incremental algorithms [2, 19]. But due to the high frequency and potentially huge information carried out in a timely fashion by data streams, these incremental approaches cannot easily handle them, unless they take the risk to make errors [18] and/or fail at estimating supports, one of the two essential components of association rules algorithms. This is where our paper takes place.

More precisely, we address the following questions:

- (a) is it possible to set up a method which replaces the *exact* support by a *statistical* support ensuring some desirable properties on support computations, and frequency estimations? Ideally, we would like the resulting support to hold regardless of the algorithm used to build or maintain frequent items/itemsets (see *e.g.* [2, 19]), and rely on mild theoretical assumptions so as to be reliably implementable.
- (b) how good is this statistical support, both from the theoretical and experimental standpoints?

The rest of this paper is organized as follows. Section 2 goes deeper into presenting the problems of dealing with uncertainty in data streams, and gives an extensive statement of our problem. Section 3 presents our solution to this problem, and its properties. Section 4 presents experimental results, and Section 5 concludes the paper with future avenues for research.

2 Problem Statement

The huge size of data streams for real-world domains compared to the limited amounts of resources to mine them makes it necessary to cope with uncertainty to achieve reasonable processing time and/or space. A significant body of previous works has addressed the accurate storing of the data stream history [1, 3, 10].

Our setting is a bit more downstream, as we question the forecasting on the data stream future. Ideally, this information is sought to be accurate not only on the data stored, but also on the whole data stream itself. For example, it's not enough to observe some item as frequent in the data stored; it is much more important to *predict* if it is really frequent in the whole data stream. Similarly, it's not enough to observe that some itemsets doesn't meet the observed frequency requirements to argue that it is *really* not frequent on the whole data stream.

From the estimation standpoint, there are two sources of error:

1. it is possible that some itemsets observed as frequent might in fact not be frequent anymore;
2. on the other hand, some itemsets observed as not frequent may well in fact be frequent from a longer history of the data stream.

Should it rely on frequencies estimations, any loss due to the imperfection of the information stored is incurred by at least one of these sources of error. The

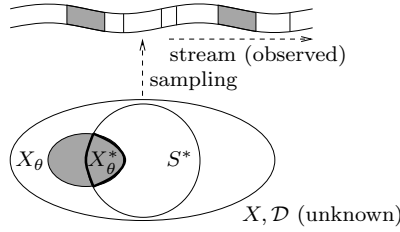


Fig. 1. Problem statement

point is that it is statistically hard to nullify both of them [17]. It is also generally impossible to capture the missing informations from the data stream to make a fully accurate prediction. Our paper is aimed at obtaining a solution to the following problem, which is a convenient relaxation of this unsatisfiable goal:

- (a) the user chooses a source of error, and fixes some related parameters;
- (b) the source of error chosen is nullified with high probability, while the other one incurs a limited loss.

It turns out that in many domains [18, 16], the relative importance of the two sources of error is not the same, and one may be much more important to control than the other one. For these domains, our approach may be a very convenient way to cope with uncertainty in finding frequent itemsets.

Now, let us skip to a slightly more formal presentation. The data stream is supposed to be obtained from the repetitive sampling of a potentially huge *domain* X which contains all possible itemsets, see Figure 1. Each itemset is sampled independently through a distribution \mathcal{D} for which we make absolutely *no* assumption, except that it remains fixed (no drift). The reader may find relevant empirical studies on concept drift for supervised mining in [5, 20]. The user specifies a real θ , the *theoretical* support, and wishes to recover all the *true* θ -frequent patterns of X . This set is called X_θ in Figure 1.

Definition 1.

$$\forall 0 \leq \theta \leq 1, X_\theta = \{T \in X : \rho_X(T) \geq \theta\} , \tag{1}$$

with $\rho_X(T) = \sum_{T' \in X: T \leq_t T'} \mathcal{D}(T')$, and $T \leq_t T'$ means that T generalizes T' .

The recovery of X_θ faces two problems. Apart from our statistical estimation problem, there is a combinatorial problem which comes from the fact that X is typically huge, even when finite. The set of observed itemsets which we have sampled from X , hereafter called S , has a size $|S| = m$ ($|S| \ll |X|$). In our framework, we usually reduce this difference with some algorithm returning a superset S^* of S , having size $|S^*| = m^* > m$. Typically, S^* contains additional generalizations of the elements of S [13]. This is not the purpose of this paper to cover this combinatorial problem; the key point is that S^* is usually still not large enough to cover X_θ , regardless of the way it is built (see Figure 1), so that the pregnancy of our statistical estimation problem remains the same.

Our statistical estimation problem can be formalized as follows:

- approximate as best as possible the following set:

$$X_{\theta}^* = X_{\theta} \cap S^* , \tag{2}$$

for any S and S^* (see Figures 1 and 2).

Remark that $\forall T \in S^*$, we cannot compute exactly $\rho_X(T)$, since we do not know X and \mathcal{D} . Rather, we have access to its best unbiased estimator $\rho_S(T)$:

$$\forall T \in S^*, \rho_S(T) = \sum_{T' \in S: T \leq_t T'} w(T') , \tag{3}$$

with $w(T')$ the weight (observed frequency) of T' in S . We adopt the following approach to solve our problem:

- find some $0 < \theta' < 1$ and approximate the set X_{θ}^* by the set of *observed* θ' -frequent of S^* , that is:

$$S_{\theta'}^* = \{T \in S^* : \rho_S(T) \geq \theta'\} . \tag{4}$$

Before computing θ' , we first turn to the formal criteria appreciating the goodness-of-fit of $S_{\theta'}^*$. The two sources of error, committed with respect to X_{θ}^* , come from the two subsets of the symmetric difference with $S_{\theta'}^*$, as presented in Figure 2. To quantify them, let us define:

$$TP = \sum_{T \in S_{\theta'}^* \cap X_{\theta}^*} \mathcal{D}(T) \tag{5}$$

$$FN = \sum_{T \in X_{\theta}^* \setminus S_{\theta'}^*} \mathcal{D}(T) \tag{7}$$

$$FP = \sum_{T \in S_{\theta'}^* \setminus X_{\theta}^*} \mathcal{D}(T) \tag{6}$$

$$TN = \sum_{T \in S^* \setminus (S_{\theta'}^* \cup X_{\theta}^*)} \mathcal{D}(T) \tag{8}$$

The *precision* allows to quantify the proportion of estimated θ -frequent that are in fact not true θ -frequents, out of $S_{\theta'}^*$:

$$P = TP / (TP + FP) . \tag{9}$$

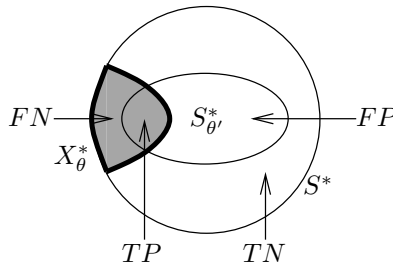


Fig. 2. The error estimation

Maximizing P leads to minimize our first source of error. Symmetrically, the *recall* allows to quantify the proportion of true θ -frequent that are missed in $S_{\theta'}^*$:

$$R = TP / (TP + FN) . \quad (10)$$

Maximizing R leads to minimize our second source of error. We also make use of a quantity in information retrieval, which is a weighted harmonic average of precision and recall, the F_β -measure. Thus, we can adjust the importance of one source of error against the other by adjusting the β value:

$$F_\beta = (1 + \beta^2)PR / (R + \beta^2P) , \quad (11)$$

A naive approach to approximate X_θ^* would typically be to fix $\theta' = \theta$. Unfortunately, the main and only interesting property of $S_{\theta'}^*$ is that it converges with probability 1 to X_θ^* as $m \rightarrow \infty$ from the Borel-Cantelli Lemma [4]. Glivenko-Cantelli's Theorem gives a rate of convergence as a function of m , but this is only useful to yield the maximization of P and R in the limit.

3 Choosing θ'

Informally, our approach boils down to picking a θ' different from θ , so as to maximize either P or R . Clearly, extremal values for θ' would do the job, but they would yield very poor values for F_β , and also be completely useless for data mining. For example, we could choose $\theta' = 0$, and would obtain $S_0^* = S^*$, and thus $R = 1$. However, in this case, we would also have $P = |X_\theta^*| / |S^*|$, a too small value for many domains and values of θ , and we would also keep all elements of S^* as true θ -frequents, a clearly huge drawback for mining issues. We could also choose $\theta' = 1$, so as to be sure to maximize P this time; however, we would also have $R = 0$, and would keep *no* element of S^* as θ -frequent. These extremal examples show the principle of our approach. Should we want to maximize the precision, we would pick a θ' larger than θ to guarantee with high probability that $P = 1$, yet while keeping large enough values for R (or F_β), and a set $S_{\theta'}^*$ not too small to contain significant informations. There is obviously a statistical barrier which prevents θ' to be too close to θ to keep the constraint $P = 1$ (Cf Section 2, last §). The objective is to be the closest to this barrier, which statistically guarantees the largest recall values under the constraint. The same principle holds for the maximization of the recall.

The following Theorem states explicitly our bound for the maximal P . Its key feature is that it holds regardless of the domain, the distribution of the itemsets, the size of S^* , or the user-fixed parameters (support, statistical risk). It relies *only* on a rather mild assumption for sampling the itemsets out of the stream.

Theorem 1. $\forall X, \forall \mathcal{D}, \forall m > 0, \forall 0 \leq \theta \leq 1, \forall 0 < \delta \leq 1$, we pick ε satisfying:

$$\varepsilon \geq \sqrt{\frac{1}{2m} \ln \frac{|S^*|}{\delta}} .$$

If we fix $\theta' = \theta + \varepsilon$ in eq. (4), then $P = 1$ with probability at least $1 - \delta$.

Theorem 2. $\forall X, \forall D, \forall m > 0, \forall 0 \leq \theta \leq 1, \forall 0 < \delta \leq 1$, we pick ε satisfying:

$$\varepsilon \geq \sqrt{\frac{1}{2m} \ln \frac{|S^*|}{\delta}} .$$

If we fix $\theta' = \theta - \varepsilon$ in eq. (4), then $R = 1$ with probability at least $1 - \delta$.

These Theorems are proven using standard tools on concentration inequalities [14]; due to the lack of space, we skip their proofs. The main point is that the values of θ' seem to be very close to the statistical barriers [17, 11] that still guarantee the maximal values for the precision or recall.

4 Experiments

We focus on evaluating how our statistical support can be helpful to mine frequent itemsets on a data stream, given a fragment of this stream. For this purpose, we use the previously defined measures: P (9), R (10) and F_β (11).

We have chosen three real life databases from the Frequent itemsets Mining Dataset Repository [9] and an association rule mining algorithm, `kdc`i [15]. The first dataset, named “Accidents” (34k transactions), holds form for each traffic accident that occurs with injured or deadly wounded casualties on a public road. The second data set, named “Retail” (88k transactions), holds customers basket from a retail supermarket store. The last dataset, named “Kosarak” (990k transactions), holds anonymized click-stream data of an on-line news portal.

To analyze the correctness of our statistical supports, we need to evaluate as many situation as possible, that is, we need to use our method with a range as large as possible for each of the free parameters. These parameters that vary during our experiments are described in Fig. 3.

Better than using a real data stream, we have chosen to simulate data streams assuming the complete knowledge of the domains, thus allowing to compute exact values for the performance measurements. More precisely, we simulate data streams by sampling each database into fragments. For example, we could consider that data arrive in a timely manner from the “Accidents” database, and that only 20% of the data is stored. So we pick 20% of the transactions of this database, we consider that it is the data stored. We have chosen to sample the database on a broad range of percentages using two scales. The first allows a fine sampling of the database, for values ranging from 1% to 10% by steps of

Database	θ	sampling1	sampling2	δ
Accidents	[.3, .9] / .05	[.01,.1] / .01	[.1, 1] / .03	[.01, .11] / .02
Retail	[.05, .1] / .01	[.01,.1] / .01	[.1, 1] / .03	[.01, .11] / .02
Kosarak	[.05,.1] / .01	[.01,.1] / .01	[.1, 1] / .03	[.01, .11] / .02

Fig. 3. Range of parameters for the experiments in the form $[a,b]/c$, where a is the starting value, c is the increment, and b is the last value

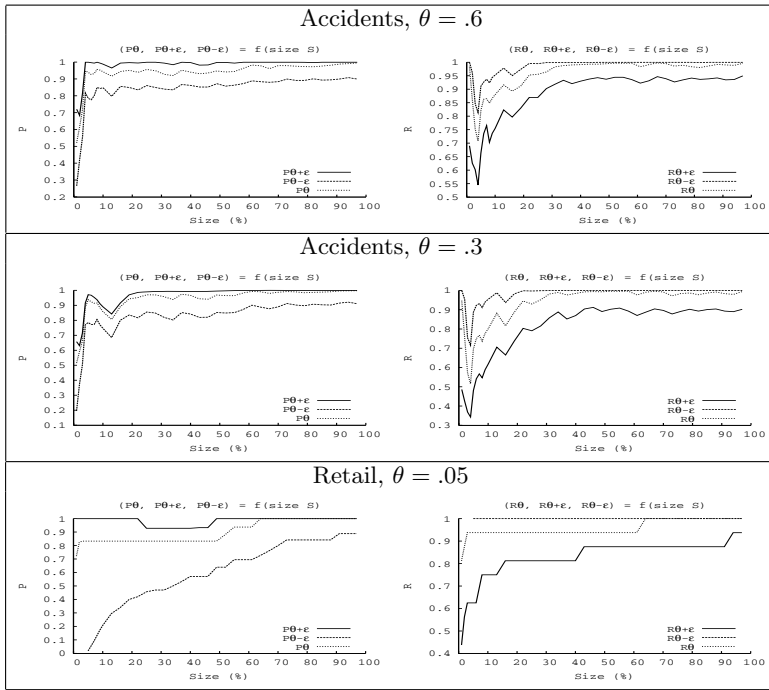


Fig. 4. Examples of plots with $\delta = .05$ and three θ values. For these values we give the P (left plot) and R (right plot) for the three methods consisting in picking $S_{\theta-\epsilon}^*$, S_{θ}^* , $S_{\theta+\epsilon}^*$

1% (“sampling1” in Fig. 3), and typically gives an idea of what may happens for very large, fast data streams. We have completed this first range with a coarse range of samplings, from 10% to 100% by steps of 3% (“sampling2”) which gives an idea of the average and limit behaviors of our method.

Finally, δ has been chosen to range through an interval of values for common statistical risks, *i.e.* from 1% to 11% by steps of 2% (see Fig. 3). Due to the very large number of experiments and the lack of space to report them all, we have put all resulting plots into web pages¹.

Figure 4 shows result from experiments on the Accidents and Retail databases with $\delta = .05$. Each plot describes for one database and one support value, either P or R of the three methods which consist in keeping $S_{\theta-\epsilon}^*$, S_{θ}^* , and $S_{\theta+\epsilon}^*$.

A first glance at these plots reveals that their behavior is almost always the same. Namely:

- the P increases with θ' (eq. 4), while the R decreases with θ' ,
- the P equals or approaches 1 for mostly storing sizes when $\theta' = \theta + \epsilon$,
- the R equals or approaches 1 for mostly storing sizes when $\theta' = \theta - \epsilon$.

¹ <http://www.univ-ag.fr/grimaag/statisticalsupports/>

These observations are in accordance with the theoretical results of Section 3. There is another phenomenon we may observe: the R associated to $\theta' = \theta + \varepsilon$ is not that far from the R of $\theta' = \theta$. Similarly, the P associated to $\theta' = \theta - \varepsilon$ is not that far from the P of $\theta' = \theta$. This shows that the maximization of P or R is obtained at a reduced degradation of the other parameter. We also remark that the P plots tend to be better than the R plots. This is not really surprising, as advocated in Section 3, since the range of values for P is smaller than that of R.

A close look at small storing sizes of the streams (before 10%) also reveals a more erratic behavior without convergence to maximal P or R. This behavior is not linked to the statistical support, but to the databases used. Indeed, small databases lead to even smaller storing sizes, and frequent itemsets kept out of small databases are in fact trickier to predict than for bigger ones. This point is important as, from a real-world standpoint, we tend to store very large databases, so we may expect this phenomenon to be reduced.

On the smallest databases, such as Retail and Kosarak, another phenomenon seems to appear. First of all, because of the small values for θ , some tests have not been performed because $\theta - \varepsilon < 0$. Furthermore, the greater difference observed between the curves seems to stem out from the different sizes of databases. For example, the Retail database is smaller than the Accidents database by a factor 3. In addition, the number of frequent itemsets found in this database is smaller than a hundred. For the sake of comparison, the Accidents database for the smallest θ gives hundreds of thousands frequent itemsets. This, we think, explains the greater differences between the curves: they are mostly a small database phenomenon, and may not be expected from larger databases.

In Figure 5, two sets of two plots taken from the Accidents database plot the F_β measure, against the size of the stream used (in %). The values of β have

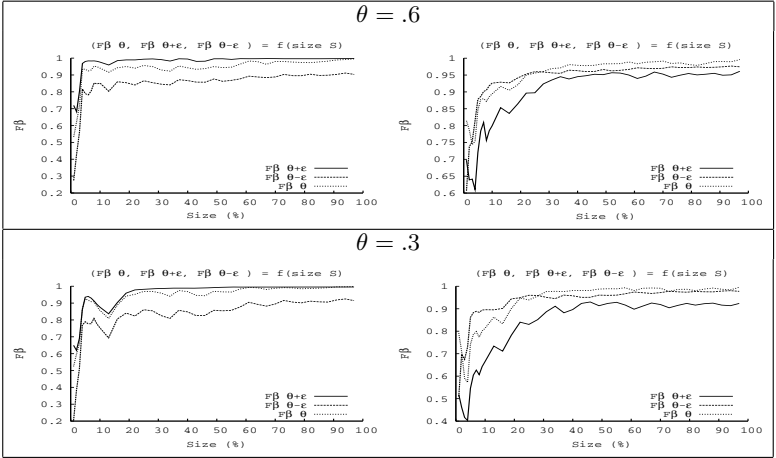


Fig. 5. Two sets of plots of the F_β value from the Accidents database, with $\beta = .2$ for the left plots and $\beta = 1.8$ for the right plots

been chosen different from 1 but not too small or too large to yield a reasonable prominence of one criterion (.2 and 1.8, see Figure 5). In each plot, the F_β value displays the advantage of choosing $\theta' = \theta \pm \varepsilon$ against the choice $\theta' = \theta$. Moreover, R that this is obtained while statistically guaranteeing the *maximal* value for whichever of P or R criterion, as chosen by the user.

5 Conclusion

One very promising research direction would be to integrate our approach with the approaches that consisting in somehow reducing the size of the data stored out of the database, so as to keep the property that itemsets *observed* as frequent still remain frequent with high probability [10]. In the framework of data streams, where we feel that such approaches take all their importance, it would be much more efficient from a statistical standpoint to keep the itemsets that are *truly* frequent (better than simply observed as frequent). This would basically boil down to mixing our approach with them, so as to keep maximal recall (this can straightforwardly be replaced by the constraint to keep maximal precision). Because of the technical machinery used in these papers (*e.g.* Blum filters [10]), mixing the approaches into a global technique for reducing the error in maintaining frequent itemsets from data streams may be more than simply interesting: it seems to be very natural.

References

1. M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. In *Proc. of the 29th International Colloquium on Automata, Languages, and Programming*, pages 693–703, 2002.
2. D. Cheung, J. Han, V. Ng, and C. Wong. Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique. In *Proc. of the 12th International Conference on Data Engineering*, pages 106–114, New Orleans, Louisiana, February 1996.
3. G. Cormode and S. Muthukrishnan. What’s hot and what’s not: Tracking most frequent items dynamically. In *Proc. of the 22nd ACM Symposium on the Principle of Database Systems*, pages 296–306. ACM Press, 2003.
4. L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.
5. W. Fan, Y.-A. Huang, H. Wang, and P.-S. Yu. Active mining of data streams. In *Proc. of the 4th SIAM International Conference on Data Mining*, pages 457–461, 2004.
6. C. Giannella, J. Han, J. Pei, X. Yan, and P.-S. Yu. *Mining Frequent Patterns in Data Streams at Multiple Time Granularities*, chapter 6. *Data Mining: Next Generation Challenges and Future Directions*. H. Karguta, A. Joshi, K. Sivakumar and Y. Yesha (Eds.). MIT/AAAI Press, 2004.
7. L. Golab and M. Tamer Ozsu. Issues in Data Stream Management. *ACM SIGMOD Record*, 2(2):5–14, June 2003.

8. S. Gollapudi and D. Sivakumar. Framework and Algorithms for Trend Analysis in Massive Temporal Data Sets. In *Proc. of the 13th International Conference on Information and Knowledge Management*, pages 168–177, 2004.
9. Frequent itemset mining dataset repository — <http://fimi.cs.helsinki.fi/data>, 2005.
10. C. Jin, W. Qian, C. Sha, J.-X. Yu, and A. Zhou. Dynamically maintaining frequent items over a data stream. In *Proc. of the 12th International Conference on Information and Knowledge Management*, pages 287–294. ACM Press, 2003.
11. M. J. Kearns and Y. Mansour. A Fast, Bottom-up Decision Tree Pruning algorithm with Near-Optimal generalization. In *Proc. of the 15th International Conference on Machine Learning*, pages 269–277, 1998.
12. G. Manku and R. Motwani. Approximate Frequency Counts over Data Streams. In *Proc. of the 28th International Conference on Very Large Databases*, pages 346–357, Hong Kong, China, 2002.
13. H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.
14. R. Nock and F. Nielsen. Statistical Region Merging. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(11):1452–1458, 2004.
15. S. Orlando, P. Palmerini, R. Perego, C. Silvestri, and F. Silvestri. kDCI: a multi-strategy algorithm for mining frequent sets. In *Proc. of the Workshop on Frequent Itemset Mining Implementations, in conjunction with ICDM 2003*, 2003.
16. S.-J. Rizvi and J.-R. Haritsa. Maintaining Data Privacy in Association Rule Mining. In *Proc. of the 28th International Conference on Very Large Databases*, pages 682–693, 2002.
17. V. Vapnik. *Statistical Learning Theory*. John Wiley, 1998.
18. A. Veloso, B. Gusmao, W. Meira, M. Carvalho, S. Parthasarathy, and M.-J. Zaki. Efficiently Mining Approximate Models of Associations in Evolving Databases. In *Proc. of the 6th European Conference on the Principles and Practice of Knowledge Discovery in Databases*, pages 435–448, 2002.
19. A. Veloso, W. Meira, M. Carvalho, B. Possas, S. Parthasarathy, and M.-J. Zaki. Mining Frequent Itemsets in Evolving Databases. In *Proc. of the 2nd SIAM International Conference on Data Mining*, pages 31–41, Arlington, April 2002.
20. H. Wang, W. Fan, P.-S. Yu, and J. Han. Mining concept-drifting data streams with ensemble classifiers. In *Proc. of the 9th International Conference on Knowledge Discovery in Databases*, pages 226–235, 2003.

Autonomous Vehicle Steering Based on Evaluative Feedback by Reinforcement Learning

Klaus-Dieter Kuhnert and Michael Krödel

University of Siegen, Institute of Real-Time Learningsystems,
Hölderlinstrasse 3, D-57068 Siegen / Germany
kuhnert@fb12.uni-siegen.de

Abstract. Steering an autonomous vehicle requires the permanent adaptation of behavior in relation to the various situations the vehicle is in. This paper describes a research which implements such adaptation and optimization based on Reinforcement Learning (RL) which in detail purely learns from evaluative feedback in contrast to instructive feedback. Convergence of the learning process has been achieved at various experimental results revealing the impact of the different RL parameters. While using RL for autonomous steering is in itself already a novelty, additional attention has been given to new proposals for post-processing and interpreting the experimental data.

1 Introduction

The study presented in this paper deals with the concept and the implementation of a system which, based on experience over a period of time, is able to autonomously learn to steer different vehicles and to optimise its behaviour to various possible road courses. This shall be done in a different way than researched in many other works before as described further below.

Key element is the fact that any action (steering, acceleration) is dependent on the situation to which a vehicle is exposed. If a vehicle is exposed to a real environment, situations are subject to permanent changes and therefore any true autonomous system will have to continuously adapt its actions.

Many research projects have been performed based on neural nets and have shown some results, but were always dependent on strong similarities between current environment and previous training pattern. A new situation always needs to be trained if deviating even slightly from previously trained situations.

The model based approach proved better success and is still being pursued in many researches. Even though we also believe in its further success, the parameterisation becomes more and more complex when the number of different situations increases (e.g. when situations are being further examined). This remains the biggest challenge for some time. In this light, an interesting variation has been proposed by using a neural net for learning the parameters of the used model [13], [14].

Altogether, however, both directions are dependent on instructive feedback – therefore they are based on a-priori knowledge resulting from parameters of teaching phases.

This paper therefore describes the research of a third method: Reinforcement Learning (RL). RL-Systems provide capabilities of self-optimising actions based on evaluative feedback. They explore the overall state-space by means of analysing the impact of previously issued actions, coping with delayed feedback as well as coping with disturbed feedback.

Given the above aspects, it should also be noted that RL is not striving to compete with the established approaches like modelling. In lieu thereof, any progress of RL-Systems might be used to enhance the advantages of modelling achieved so far. At the end, a combined system built on modelling and RL might provide better results than each approach alone. In this light, we strongly believe RL-system will play a significant role in the near future in autonomous driving systems.

This paper, however, focuses purely on Reinforcement Learning in order to explore its benefits and limitations.

All in all, the main targets of this research are: steering of an autonomous vehicle along any curvy road, autonomous exploration of new actions for familiar as well as for new situations, therefore autonomous optimization (self-tuning of the system to any combination of environment and vehicle), learning from evaluative feedback (in contrast to instructive feedback/ teaching), coping with delayed feedback (delayed rewarding) as well as non-linearity of true environment, and finally Real-time processing.

2 Related Work

Till now the visual control of systems for autonomous vehicle driving with learning components have been implemented in several ways. [2] describes a short direct connection between image processing and soft computing learning method using a neural network. This approach provides good results but only as long as input pictures of the scene are similar to the training patterns. This approach was being enhanced by a multiple neural network [3], but could not completely solve the dependency problem of the taught training patterns. Further developments then included a GPS system [4] to support orientation or enhanced the approach with object-oriented vision in order to distinguish between road following and obstacle detection [5], [6]. In all those variations, however, neural networks with their inherent dependency on training patterns are embedded. Also, as a major difference to the presented research, the established knowledge on vehicle driving is stored within the neural net but not explicitly available, e.g. for optimisation or for further learning processes.

A completely different approach is being followed by using explicit modelling, therefore trying to rebuild a model of the environment as well as the vehicle and to derive proper actions from it. The basic idea of such a model is to try to understand interaction between vehicle and environment and to predict consequences of any behaviour thus allowing to determine a suitable behaviour in a given situation.

The major challenge of this approach is to find a suitable model which approximates the true vehicle behaviour and environment in the best way. Any difference between the model and the real environment/vehicle results in a difference between the calculated behaviour and an optimum behaviour. Any model also needs to be shaped up and tuned with parameters. Usually there are no versatile models, so

any change of e.g. vehicle or environment requires a corresponding tuning, respectively adaptation of the model. In other words, any tuned model is valid only for a certain environment or vehicle and is more or less sensible to any change of these. [7] describes an early success with international attention of a vehicle system using a real-time vision system BVV2 [8]. Further developments in this area (e.g. [9]) are being pursued with significant progress, however always dependent on many parameters for the modelling process.

3 General Structure

Figure 1 shows the overall structure of our approach. According to Sutton/Barto [1], a RL-system consists of an RL-Agent and the RL-Environment. The RL-Agent receives input regarding the state (situation) s_t as well as a reward r_t and determines an appropriate action a_t . This action will cause a reaction of the RL-Environment and consequently result in a change of state from s_t to s_{t+1} . Similarly, the RL-Environment will also issue a reward r_{t+1} corresponding to s_{t+1} .

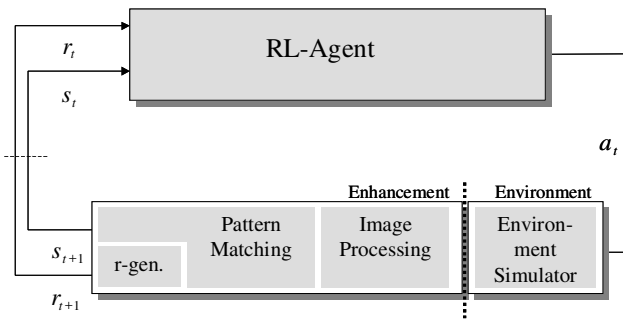


Fig. 1. Structure of the System

Since the determination of the state s and the reward r is required from the RL-Environment and usually not being provided by an environment simulator, our system enhances the RL-Environment and provides methods of Image Processing, Pattern Matching and reward-generation being described more in detail in the next paragraph.

4 Image Processing and Pattern Matching

The proper pre-processing of the incoming data is key to any successful RL-System. One of the major novelties of this research is the determination of a suitable situation description in correspondence to the situation the vehicle is in. In this light, the classical RL-Environment (the lower part of figure 1) has been enhanced in order to provide the RL-agent with defined descriptions of each situation. Any incoming image, along with external information on any appropriate action, is being given to an image processing system, which extracts all relevant information in order to describe

the current situation. Such situation description is being referred to as Abstract Complete Situation Description (ACSD). Even though such technique is a significant part of the current research, it shall not be described at this point since public presentations (also available on the web) have been done and has been described in proceeding papers in detail (e.g. [10], [11], [12]). At this point it shall only be emphasized that it makes use of a self-created statistical database storing the conditional probabilities of road mark positions and additionally exploiting the information to extract the road marks faster and more reliable than with many other methods. Such ACSDs are then being stored along with the corresponding action a of the training phase in a database.

When operating the system in driving mode, any incoming image is being converted into an ACSD. Given the ACSD of the current image and the ACSD's in the database, a fast k-nearest neighbour algorithm locates the most similar ACSDs. Such way, the RL-Agent not only receives information regarding the current situation but also information which other similar (or identical) situations experienced before.

In this context, the ACSD explained above is being used as the state s , the action a is basically the steering command (i.e. angle of the steering wheel).

Additionally, a reward r is being determined, which can be a value representing the lateral deviation from the middle of the road if the agent has to learn to follow any road course – however, the reward can also be a timer value measuring the time needed for a road section if the agent is to learn to pass a road section in the shortest possible time.

5 Reinforcement Learning

The basic idea of RL is, that states s_t , respectively actions issued at a certain state a_t , are being rated considering the reward r_{t+1} of the situation s_{t+1} . Such rating is being represented by $Q(s,a)$ and is defined to be the sum of future rewards discounted by a discount factor γ . In the beginning only estimates of the q-values exist. Thus, the current Q-values deviate from the converged Q-values by an error TDerr.

$$Q(s_t, a_t) = r_{t+1} + \sum_{i=2}^{\infty} \gamma^i r_{t+i} - TDerr \quad (1)$$

$$Q(s_t, a_t) = r_{t+1} + \gamma \cdot Q(s_{t+1}, a_{t+1}) - TDerr \quad (2)$$

$$TDerr = r_{t+1} + \gamma \cdot Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \quad (3)$$

The error TDerr is being used for updating the Q-values (also discounted by the learning rate parameter α) and will lead to a convergence of the Q-values, as long as the reward is deterministic.

$$Q(s_t, a_t) := Q(s_t, a_t) + \alpha \cdot TDerr \quad (4)$$

The maximum Q-value of a state s across all possible actions shall be:

$$Q_{a-\max} = \max_a Q(s_i, a) \tag{5}$$

and in combination with the policy π , the system usually selects the action with the highest Q-value, resulting the system to operate in the mode called exploitation mode:

$$\pi(s_i) : a = a(Q_{a-\max}) \tag{6}$$

An initial target, however, is to self-optimize behaviour over time. Consequently it is imperative to actively explore the state-action-space in order to search for the best action (and temporarily switching to exploration mode):

$$\pi(s_i) : \begin{cases} a = a(Q_{a-\max(i)}) \text{ for } \text{rand}() \geq \epsilon \\ a = \text{rand}() \text{ else} \end{cases} \tag{7}$$

with $\text{rand}() \in [0,1], \epsilon \in [0,1]$

In this policy learning and exploitation are randomly mixed. Such way the RL-system also adapts autonomously to a new or even changing environment without any explicit training phase.

Notable, at this point, is also the capability of a RL-system to cope with non-linearity's (e.g. errand rewarding) of the environment. This notation also includes the ability of the system to cope with delayed rewards. Given is for example the state-action relationship as displayed in figure 2. At $t=0$ the system shall be in state s_i and has the option between two actions: $a_{i \rightarrow j}$ which will cause a transition to state s_j and further-on to state s_k or action $a_{i \rightarrow i}$ which will prevent any change of state. r_j shall be errand and r_k shall be much higher than r_i . Therefore, the system should be able to learn to accept an errand (low) temporary reward at state s_j but to finally reach s_k and should not remain in state s_i .

According to its policy, the system will choose the action with the highest Q-Values. Depending on the rewards and the discount factor, the maximum Q-value at state s_i is given in formula 8. Basically, the closer the discount value get towards "1", the higher will be the preference on long-term-rewards instead of short-term rewards.

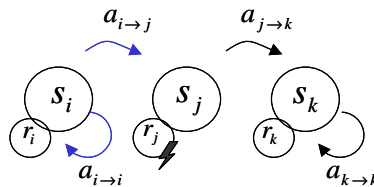


Fig. 2. State sequence of the RL-system while coping with disturbed reward r_j (see flash)

$$Q_{a-\max(i)} = \max \left\{ \frac{r_i}{1-\gamma}; r_j + \gamma \frac{r_k}{1-\gamma} \right\} \tag{8}$$

6 Experimental Results and Findings

6.1 Experimental Setup

The experiments have been done with a closed-loop-system consisting of two connected computers. The System-PC, processes the video stream of a connected camera and calculates the steering commands for the vehicle. These steering commands are then being given to the Simulator-PC, which is responsible for the environment simulation. A converter box connects both interfaces. The output of the second computer is being given onto its monitor, which again is being read by the video camera – alternatively, the video output of the Simulator-PC is being connected directly to the framegrabber of the system-PC using a S-VHS-cable. Due to this set-up a realistic amount of measurement noise is introduced into the system.

The task of the following experiments has been, to learn the ideal vertical position, respectively, the driving angle of a vehicle driving along a road course. In this light, a simplified system with 11 possible steering commands (equally distributed from sharp left steering to sharp right steering) has been defined. The number of possible situations varies depending on the settings of the image processing part.

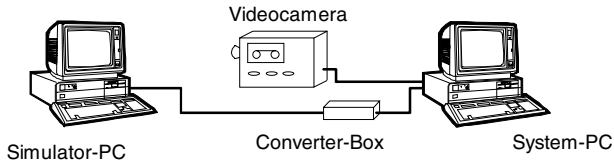


Fig. 3. HW setup for this research using two interconnected computers

At this point it should be noted that all further results have been achieved without any supervised teaching at all! Therefore, the system discovers the whole state space completely on its own – in detail: the appropriateness of every action of every situation. Such extreme exploration of the environment is only possible on a simulator, which is our main reason for choosing such a platform.

6.2 Splining and Measurement of Convergence

One of the major new and significant findings in this research was, that a criterion is needed as to how much the system converged. Even though the values of $TDerr$ (formula 3) represent the convergence error and is therefore the basis for the update, the chart of $TDerr(t)$ does not express the grade of state-space convergence. Fig. 4 shows lateral converged state-space (optimal lateral position to be learned) after the issuance of approx. 170.000 actions and a chart of $TDerr$ over time – the convergence is not really be recognizable.

Regarding the state-space: the state “R1” is equivalent to being at the left edge of the road; the best action in this situation is the selection of the center situation. The state “R11” is equivalent to being at the right road edge; the best action in this

situation is again the selection of the center situation. The action “1” is equivalent to selecting the leftmost situation, the action “11” is equivalent to selecting the rightmost situation.

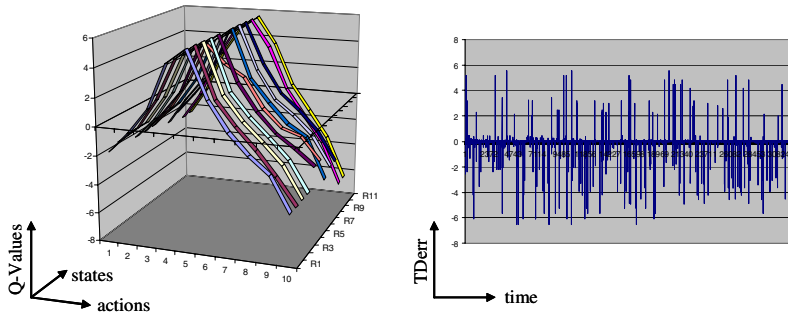


Fig. 4. Original state-space (approx 30.000 actions) and TDerr (t)

Therefore the state-space of the Q-values is being approximated by calculating for all situations each one spline over all actions. The cumulated squared difference between all original Q-Values and it's corresponding splined Q-Value results in the determination of a value “Smoothness”. The splined Q-Values as well as the development of the Smoothness Value during the same testseries as Fig. 4 is shown in Fig. 5 and a clear indication for convergence can be seen. A rising smoothness value indicates the adaptation of the system to its environment (major learning performed), because the action space has to be globally smooth for the chosen system. The smoothness value decreases while the system converges. A complete convergence (therefore smoothness-value equal to zero) will not be achieved since any true environment is also not absolutely deterministic.

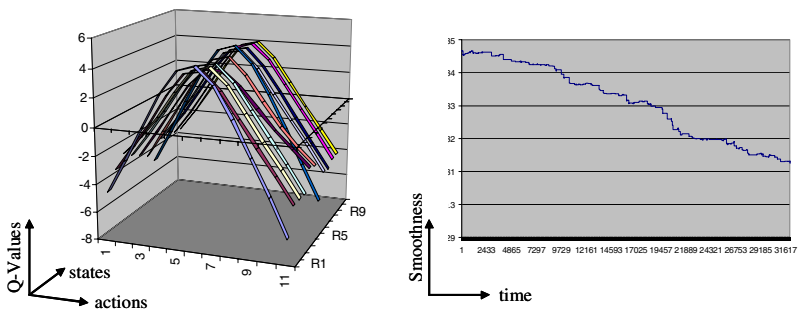


Fig. 5. Splined state-space (approx 30.000 actions) and Smoothness(t)

However, a disadvantage of the splining, is the distortion of the state-space if some actions did not get issued often enough – resulting in a less often update according to Reinforcement Learning. Fig. 6 shows the original Q-Values as well as the splined Q-

Values for a state-space, in which only the middle actions got issued often enough (resulting in a local convergence). As a solution to this dilemma, the number of updates for each Q-Value gets counted and can either be considered during the splining process or used to hide the associated Q-Values.

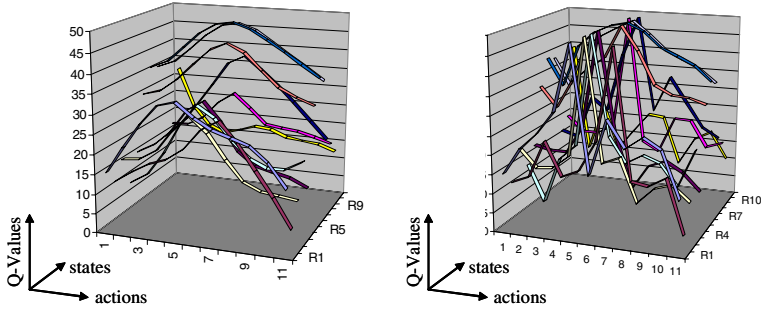


Fig. 6. Original state-space and splined state-space

Fig. 6 also shows the impact of reduced exploration. At reduced exploration, some actions might nearly never get issued (since not any action can be issued from any situation, creating a situation-specific dependency). Partially, this can be overcome by longer test-cycles but still, the counted number of updates for each Q-Values needs to be considered for any further analysis.

6.3 Impact of Learning Parameters

Although some other publications deal with the combination of RL and autonomous driving, the impact of the RL parameters are not yet publicly documented. In consequence, quite some experiments have been spent on such topic and provide for the first time an overview of the impact of the basi RL-parameters. Regarding the learning rate parameter α on the learning process, Fig. 7 and Fig. 8 show two similar testseries which differ only in values of α . A small value for α results in slower, but more stable learning. It should be noted that for those and the further tests, the system had to learn the driving angle, ie. the optimal driving angle is dependant on the situation the vehicle is in resulting in a different position of the maximum Q-Value for each situation.

An environment with a higher number of situations lead to a more complex state space. Fig. 9. show corresponding tests; again with different settings for the grade of exploration. All in all, the system performs the learning task quite well – especially, as mentioned above, without any teaching at all. The more complex the environment becomes (dimension of state-space increasing) the test duration needs to be enhanced accordingly. However, even extended test times might run into limitations when the environment gets more and more complex. In those cases, focused exploration (i.e. exploration of only some sub-areas of the whole state-space) are supposed to be a viable solution – further investigation on this matter is planned for the near future.

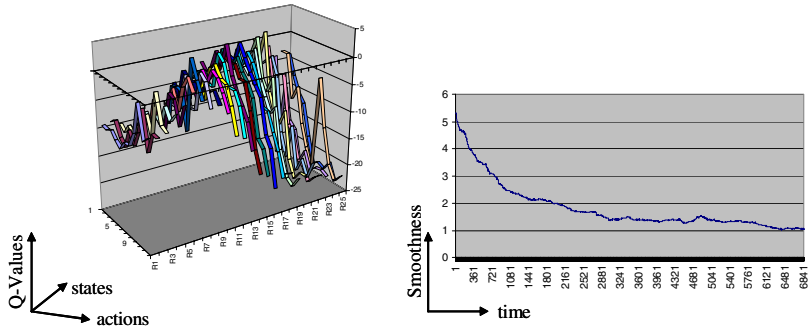


Fig. 7. Testseries (approx 7.000 actions) with $\alpha = 0.1$; original state-space and Smoothness(t)

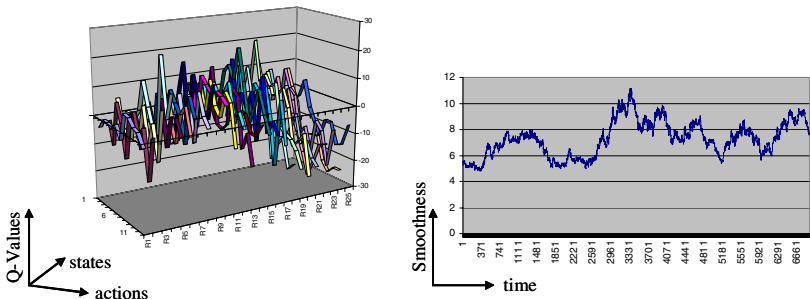


Fig. 8. Testseries (approx. 7.000 actions) with $\alpha = 0.5$; original state-space and Smoothness(t)

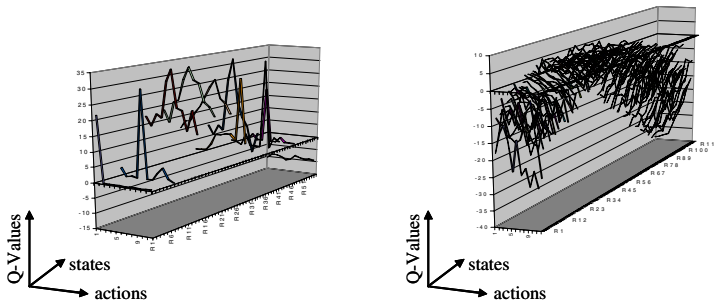


Fig. 9. Impact of exploration: $\epsilon = 0,1$ (left) resp. $\epsilon = 1,0$ (right)

7 Summary

Pattern Matching provides capabilities of autonomous driving with knowledge being directly accessible (for further optimization). In addition, Reinforcement Learning allows autonomous optimization of behaviors based on self-created rewards, even if delayed or disturbed. Combining both techniques allows learning and optimizing of

visual steering of autonomous vehicles. The current research, will now be further used in more complex environments in order to explore the limitations of exploration in combination to test duration. Also, further aspects regarding coping with delayed rewards will still be focussed on within the current research.

References

- [1] Richard Sutton, A. G. Barto, Reinforcement Learning: An introduction, MIT-Press, 2000, Cambridge (USA)
- [2] D. A. Pommerleau, Efficient Training of Artificial Neural Networks for Autonomous Navigation, *Neural Computation* 3, 1991
- [3] T.M. Jochem, D.A. Pomerleau, C.E. Thorpe. MANIAC: A Next Generation Neurally Based Autonomous Road Follower, IAS-3, Int. Conference on Intelligent autonomous Systems, February 15-18, 1993, Pittsburgh/PA, USA, F.C.A. Groen, S.Hirose, C.E.Thorpe (eds), IOS Press, Washington, Oxford, Amsterdam, Tokyo, 1993
- [4] T.M.Jochem, D.A.Pomerleau, C.E.Thorpe, Vision Guided Lane Transition, *Intelligent Vehicles '95 Symposium*, September 25-26, 1995, Detroit/MI, USA
- [5] S.Baluja, D.A.Pomerleau, Expectation-based selective attention for visual monitoring and control of a robot vehicle, *Robotics and Autonomous System*, Vol.22, No.3-4, December, 1997
- [6] Uwe Franke, Dariu Gavrilla, Steffen Görzig, Frank Lindner, Frank Paetzold, Christian Wöhler, *Autonomous Driving Goes Downtown*, *IEEE Intelligent Vehicles Systems*, v.13 n.6, p.40-48, November 1998
- [7] E.D.Dickmanns, A.Zapp, Autonomous High Speed Road Vehicle Guidance by Computer Vision, *Preprints of the 10th World Congress on Automatic Control*, Vol.4, International Federation of Automatic Control, Munich, Germany, July 27-31, 1987
- [8] K.-D.-Kuhnert, A Vision System for Real Time Road and Object Recognition for Vehicle Guidance, *Proc. Mobile Robots*, Oct 30-31, 1986, Cambridge, Massachusetts, Society of Photo-Optical Instrumentation Engineers, SPIE Volume 727
- [9] E.D.Dickmanns, R.Behringer, D.Dickmanns, T.Hildebrandt, M.Maurer, F.Thomanek, J.Schiehlen, The Seeing Passenger Car 'VaMoRs-P', *Intelligent Vehicles '94 Symposium*, October 24-26, 1994, Paris, France
- [10] M. Krödel, K.-D. Kuhnert, Pattern Matching as the Nucleus for either Autonomous Driving or Drive Assistance Systems, *IEEE Intelligent Vehicle Symposium*, June 17-21, 2002, Versailles, France
- [11] K.-D. Kuhnert, M. Krödel, Reinforcement Learning to drive a car by pattern matching, *Annual symposium of Pattern recognition of DAGM*, September 16-18, 2002, Zurich (Switzerland)
- [12] K.-D. Kuhnert, M. Krödel, Autonomous Driving by Pattern Matching and Reinforcement Learning, *International Colloquium on Autonomous and Mobile Systems*, June 25-26, 2002, Magdeburg, Germany
- [13] K.-D. Kuhnert, W. Dong, Über die lernende Regelung autonomer Fahrzeuge mit neuronalen Netzen, 18. Fachgespräch Autonome Mobile Systeme (AMS), December 4-5, Karlsruhe, Germany
- [14] W. Dong, K.-D. Kuhnert, Robust adaptive control of nonholonomic mobile robot with parameter and non-parameter uncertainties, *IEEE Transaction on Robotics and Automation*, 2004

Cost Integration in Multi-step Viewpoint Selection for Object Recognition

Christian Derichs^{1,*}, Frank Deinzer², and Heinrich Niemann¹

¹ Chair for Pattern Recognition, Department of Computer Science,
Universität Erlangen-Nürnberg, Martensstr. 3, 91058 Erlangen
{derichs, niemann}@informatik.uni-erlangen.de

² Siemens Medical Solutions, Siemensstr. 1, 91301 Forchheim
frank.deinzer@siemens.com

Abstract. During the last years, computer vision tasks like object recognition and localization were rapidly expanded from passive solution approaches to active ones, that is to execute a viewpoint selection algorithm in order to acquire just the most significant views of an arbitrary object. Although fusion of multiple views can already be done reliably, planning is still limited to gathering the next best view, normally the one providing the highest immediate gain in information.

In this paper, we show how to perform a generally more intelligent, long-run optimized sequence of actions by linking them with costs. Therefore it will be introduced how to acquire the cost of an appropriate dimensionality in a non-empirical way while still leaving the determination of the system's basic behavior to the user.

Since this planning process is accomplished by an underlying machine learning technique, we also point out the ease of adjusting these to the expanded task and show why to use a multi-step approach for doing so.

Keywords: Viewpoint Selection, Active Vision, Reinforcement Learning.

1 Introduction

In the last few years, a lot of approaches have been made to add an active component to the task of combined object recognition and localization, leading to algorithms teaching an agent which views of an arbitrary object to take. Therefore, most of them apply quite powerful machine learning techniques to this task, like Reinforcement Learning [7], but still restrict themselves to just performing a single-step process, that is to always select the immediate best view to be taken within the next step for increasing the classification rate. Obviously this behavior might be suboptimal concerning the effort needed for reaching a certain classification reliability, e.g. if the agent passes by a viewpoint in a first step it has to take later on anyway.

* This work was funded by the German Science Foundation(DFG) under grant SFB 603/TP B2. Only the authors are responsible for the content.

In light of this drawback, the work of this paper will show how to put a more foresighted component to the viewpoint selection problem by integrating costs to the system. While further cost-integrating approaches attempt to control the effort feasible for improving the classification itself [5] or to teach the agent by introducing costs for misclassification [4], we do explicitly not aspire at an enhancement in classification. Moreover our goal is to gain similar results, but with possibly less effort. Therefore the components of an agent's possible action have to be weighted against each other by introducing those actions' costs, respectively cost relations, which in turn have to be adjusted to the learning technique's reward modeling. Unlike [6], where those costs are determined empirically, our attention is towards acquiring them as a by-product of the training performed anyway, namely by learning and directly applying them in parallel manner to the agent-teaching, main learning process.

Thus section 2 will give a summary of the machine learning technique, namely Reinforcement Learning, underlying the viewpoint selection task. It will be directly introduced as a multi-step approach since our presented way of cost integration makes fundamental use of this assumption. As the basics are defined then, section 3 provides the method for establishing and integrating cost-factors to the already existing learning algorithms, while section 4 will introduce methods for computing the new target values more reliably and preferably without extending the processing time of the global learning process. Results concerning a cost-sensitive solution of the viewpoint selection problem will finally be shown for a co-operative environment where classification itself does not make up a critical task.

2 Viewpoint Selection Without Costs

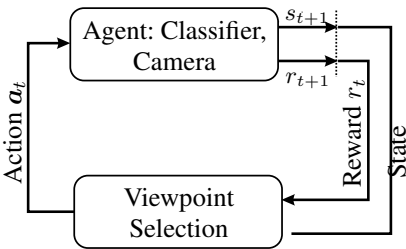


Fig. 1. Principles of Reinforcement Learning applied to viewpoint selection

The goal of this work is to provide a solution to the problem of optimal viewpoint selection for 3D object recognition without making a priori assumptions about the objects and the classifier. The problem is to determine the next view of an object given a series of previous decisions and observations and can also be seen as the determination of a function which maps a history of observations to a new viewpoint. This function should be estimated automatically during a training step and should improve over time. Additionally,

the function should take uncertainty into account in the recognition process as well as in the viewpoint selection, be classifier independent and be able to handle continuous viewpoints.

A straightforward and intuitive way to formalizing the problem is given in fig.1. A closed loop between sensing state s_t and performing action \mathbf{a}_t can be seen. The chosen actions \mathbf{a}_t

$$\mathbf{a}_t = (a_{t,1}, a_{t,2}, \dots, a_{t,n})^T \quad (1)$$

correspond to the movement of the camera at time t .

The sensed states s_t , actually probability densities (2), are estimated by the employed classifier. In this paper we use a classifier [2, 3] that is able to perform a fusion of multiple acquired images. In active object recognition, a series of observed images $\mathbf{f}_t, \mathbf{f}_{t-1}, \dots, \mathbf{f}_0$ of an object are given together with the camera movements $\mathbf{a}_{t-1}, \dots, \mathbf{a}_0$ between these images. Based on these observations of images and movements one wants to draw conclusions for a non-observable state \mathbf{q}_t of the object. This state of the object consists of its class and pose relative to the camera. In the context of a Bayesian approach, the knowledge on the object's state is given in form of the a posteriori density $p(\mathbf{q}_t | \mathbf{f}_t, \mathbf{a}_{t-1}, \mathbf{f}_{t-1}, \dots, \mathbf{a}_0, \mathbf{f}_0)$ that is calculated by the classifier. In [2] it is discussed how to represent and evaluate this density with particle filter approaches. From the viewpoint selection's perspective, this density contains all the necessary information, so that the following *density* definition meets the Reinforcement Learning requirements:

$$s_t = p(\mathbf{q}_t | \mathbf{f}_t, \mathbf{a}_{t-1}, \mathbf{f}_{t-1}, \dots, \mathbf{a}_0, \mathbf{f}_0) \quad (2)$$

Additionally, the classifier returns a so called *reward* r_t , which measures the quality of the chosen viewpoint. For a viewpoint that increases the information observed so far, the reward should have a large value. A well-know measure for expressing the informational content that fits our requirements is the entropy

$$r_t = -H(s_t) = -H(p(\mathbf{q}_t | \mathbf{f}_t, \mathbf{a}_{t-1}, \mathbf{f}_{t-1}, \dots, \mathbf{a}_0, \mathbf{f}_0)) \quad . \quad (3)$$

It is important to notice that the reward should also include costs for the camera movement and object classification. This topic will be discussed in section 3.

At time t during the decision process, i.e. the selection of a sequence of viewpoints, the goal will be to maximize the accumulated and weighted future rewards, called the *return*

$$R_t = \sum_{n=0}^{\infty} \gamma^n r_{t+n+1} = - \sum_{n=0}^{\infty} \gamma^n H(s_{t+n+1}) \quad \text{with } \gamma \in [0; 1]. \quad (4)$$

The weight γ defines how much influence a future reward at time $t + n + 1$ will have on the overall return R_t . So, to meet the demands of this paper, a $\gamma > 0$ is required since we make use of a multi-step approach. Of course, the future rewards cannot be observed at time step t . Thus, the following function, called the *action-value function* $Q(s, \mathbf{a})$

$$Q(s, \mathbf{a}) = E \{ R_t | s_t = s, \mathbf{a}_t = \mathbf{a} \} \quad (5)$$

is defined, which describes the expected return when starting at time step t in state s with action \mathbf{a} . In other words, the function $Q(s, \mathbf{a})$ models the expected quality of the chosen camera movement \mathbf{a} for the future, if the classifier has returned s before.

Viewpoint selection can now be defined as a two step approach: First, estimate the function $Q(s, \mathbf{a})$ during training. Second, if at any time the classifier returns s as result, select that camera movement which maximizes the expected accumulated and weighted rewards. This function is called the *policy*

$$\pi(s) = \operatorname{argmax}_{\mathbf{a}} Q(s, \mathbf{a}) \quad . \quad (6)$$

The key issue of course is the estimation of the function $Q(s, \mathbf{a})$, which is the basis for the decision process in (6). One of our demands is that the selection of the most

promising view should be learned without user interaction. Reinforcement Learning provides many different algorithms to estimate the action value function based on a trial and error method [8]. Trial and error means that the system itself is responsible for trying certain actions in a certain state. The result of such a trial is then used to update $Q(\cdot, \cdot)$ and to improve its policy π .

In Reinforcement Learning, a series of *episodes* are performed: Each episode k consists of a sequence of state/action pairs $(s_t^k, \mathbf{a}_t^k), t \in \{0, 1, \dots, T\}$, with T steps at most. Each performed action \mathbf{a}_t in state s_t results in a new state s_{t+1} . During the episode, new returns $R_t^{(k)}$ are collected for those state/action pairs (s_t^k, \mathbf{a}_t^k) which have been visited at time t during the episode k . At the end of the episode the action-value function is updated. In our case, so called Monte Carlo Learning [8] is applied and the function $Q(\cdot, \cdot)$ is estimated by the mean of all collected returns $R_t^{(i)}$ for the state/action pair (s, \mathbf{a}) for all episodes.

As a result for the next episode one gets a new decision rule π_{k+1} , which is now computed by maximizing the updated action value function. This procedure is repeated until π_{k+1} converges to the optimal policy. The reader is referred to a detailed introduction to Reinforcement Learning [8] for a description of other ways for estimating the function $Q(\cdot, \cdot)$. Convergence proofs for several algorithms can be found in [1].

Since most of the algorithms in Reinforcement Learning treat the states and actions as discrete variables, a way to extend the algorithms to continuous Reinforcement Learning is to approximate the action-value function $\widehat{Q}(s, \mathbf{a})$ which can be evaluated for any continuous state/action pair (s, \mathbf{a}) . Basically, this approximation is a weighted sum of the action-values $Q(s', \mathbf{a}')$ of all previously collected state/action pairs. All the collected action-values $Q(s', \mathbf{a}')$ are referred as Q -base throughout this paper. For the exact calculation of $\widehat{Q}(s, \mathbf{a})$ please refer to [3].

Viewpoint selection, i.e. the computation of the policy π , can now be written, according to (6), as the optimization problem

$$\pi(s) = \underset{\mathbf{a}}{\operatorname{argmax}} \widehat{Q}(s, \mathbf{a}). \quad (7)$$

This problem (7) can be solved by global optimization techniques. In this work we use the Adaptive Random Search algorithm (ARS) [9] combined with a local simplex.

3 Integration of Costs

Referring to the previous sections we first have to integrate the costs of an action into the reward of Reinforcement Learning. This is done by adding a cost term $C(\mathbf{a}_t)$ to the reward in (3) yielding

$$r_{t+1} = -H(s_{t+1}) - C(\mathbf{a}_t) \quad (8)$$

It is now necessary to take a closer look at the detailed modeling of those costs to be able to understand its influence on the system-determining reward and its interaction with $H(s_{t+1})$. In general, the description is

$$C(\mathbf{a}_t) = \sum_i \kappa_i \cdot x_{t,i} = \boldsymbol{\kappa} \cdot \mathbf{x}_t \quad \text{with} \quad x_{t,i} = \frac{a_{t,i}}{a_{i_u}} \quad (9)$$

where index i provides a handle to the i th element of an action \mathbf{a} an agent is actually able to perform within a given environment. Next to this, every action component $\mathbf{a}_{t,i}$ can be split into an amount of $x_{t,i}$ unit actions a_{i_u} for every time step. Latter ones have to be linked with a specific cost-factor κ_i in order to determine (9).

While now a_{i_u} as well as κ_i can take influence on the cost of an action we decided to let the user choose all a_{i_u} in a meaningful way beforehand, whereas κ_i will be the crucial, variable parameter for affecting the system behavior via $C(\mathbf{a}_t)$. Here the user is asked for i values of absolute type which might be difficult to acquire in a real world environment since they might be influenced by various conditions. But what is essential to be provided is at least a constant relation $\frac{\kappa_i}{\kappa_j}$ between all involved cost-factors, which are likely to be achieved appropriately, as they might compare intuitive aspects like the time two different unit action components will use to execute or the energy use of moving compared to that of zooming, for example. Doing so, (9) is no longer dependent on κ , but just on a single κ_i , typically κ_1 , and the given relations m_i .

$$C(\mathbf{a}_t) = \kappa_1 \cdot \sum_i m_i \cdot x_{t,i} \quad ; \quad m_i = \frac{\kappa_i}{\kappa_1} \quad (10)$$

The problem occurring here is the theoretically unlimited range of κ_1 combined with its independency of $H(s_{t+1})$ in general. So just choosing the first cost-factor randomly or by sense of proportion would mostly lead to an imbalance between the terms of the sum in (8), that is the system is either almost exclusively influenced by the curve of $-H(s_{t+1})$ or just trying to reduce added costs without regarding the entropy based reward. While the user might be able to choose a sufficient value for κ_1 in a simple environment beforehand, he will surely fail when actions and rewards become more complex. So the only way, apart from gaining it empirically, is to learn a cost-factor fitting the entropy based reward's order of magnitude and then integrate it into the Reinforcement Learning process.

Consequently, we have to introduce a criterion for rating different κ_1 while naturally not integrating κ_1 into this criterion directly. Regarding our demands we are eager for a cost-factor leading the agent to acquire an entropy based reward as high as possible on the one hand, while causing as little cost as possible on the other. To be more precise, we do not seek to reduce the costs themselves but the mixed and weighted amount of the various cost generating unit actions a_{i_u} . Doing so, the κ_1 under testing has no direct influence on its own rating. It just affects the decision process of the Reinforcement Learning regarding (10), resulting in eventually different actions to be taken. So using a rating function

$$f(\kappa_1) = \frac{-H(s_{t+1})}{\sum_i x_{t,i} \cdot m_i} = \frac{-H(s_{t+1})}{\frac{C(\mathbf{a}_t)}{\kappa_1}} \quad (11)$$

such a change in behavior will affect the denominator directly and usually also the nominator, leading to a different rating for the appropriate κ_1 .

While (11) is laid out for a single-step task, the intention of this paper requires expanding it to work with multi-step problems up to episode length T . Therefore using the information content's difference of consecutive views $\Delta(-H(s_{t+1})) = -H(s_{t+1}) + H(s_t)$, the equation is changed to

$$\tilde{f}(\kappa_1) = \frac{\sum_{t=0}^{T-1} \Delta(-H(s_{t+1}))}{\sum_{t=0}^{T-1} \sum_{i=1}^n x_{t,i} \cdot m_i} = \frac{\sum_{t=0}^{T-1} \Delta(-H(s_{t+1}))}{\sum_{t=0}^{T-1} \frac{C(\mathbf{a}_t)}{\kappa_1}} = \text{CBR} \quad . \quad (12)$$

At this point, it becomes obvious why to use a multi-step approach. Instead of calculating a conjoint, single sum of the nominator and denominator in (12) for each time step this is done separately since we would like to find one κ_1 that maximizes the nominator over an entire episode while minimizing the denominator over an entire episode at the same time. So actually we do not care about the ratings of every single time step, what a combined sum would imply. Furthermore, matching this demand we need to provide a multi-step approach for Reinforcement Learning with $\gamma > 0$, because otherwise calculating such a conjoint sum would always result in the same rating as in (12).

When finally applying each κ_1 to be tested to several episodes within the training phase of Reinforcement Learning, the one resulting in the highest values for (12) on average is regarded to be the optimal one. A further assumption made in our approach is the permanent existence of an action component $a_{t,n}$ in (1) related to taking views. This turns out to be quite useful since our demand always is to solve the task with the minimal amount of views if additional costs allow to do so. Moreover (12) will be called the CBR for cost-benefit ratio in the following due to its appearance.

4 Learning an Appropriate Cost-Factor

So far, the two main issues of our viewpoint selection approach were described. On the one hand, it is necessary to learn a cost-factor κ_1 , and on the other hand a regular improvement of the Q -base, influenced by the costs, is needed. Therefore, a consequential first thought is to just perform a whole training phase with a fixed κ_1 and to repeat this for all cost-factors one would like to examine. Then comparison could simply be done by using (12) in a global way, i.e. by calculating the average of all episodes' CBR. As this process would result in a maximal CBR for a certain κ_1 , it is just a question of iterations to get closer to the global maximum. Although this would work for sure, the drawback is obvious since we would have to perform a multiple of training episodes, which becomes unbearable soon. For this reason, the following three subchapters will introduce mainly independent approaches for keeping the amount of necessary episodes within the training phase as small as possible.

4.1 Expanding the Q -Base

So, a mostly parallel procedure for κ - and Reinforcement Learning had to be found while handling the occurring interdependencies at the same time. Regarding this, the main problem turned out to be the fact that a $Q(s', \mathbf{a}')$ learned with κ_{1x} usually does not provide a good strategy when using κ_{1y} . Thus already executed episodes might store information in the Q -base that is completely useless or even harmful for later evaluation if κ_1 is completely different. This consequently results in a lot of episodes executed for nothing and finally missing for providing a more reliable set of reference values.

To reduce this drawback in a first instance an expansion was added to all the $Q(s', \mathbf{a}')$ in the Q -base. While so far just storing state s_t , action \mathbf{a}_t and $Q(s', \mathbf{a}')$ itself, additionally all actions \mathbf{a}_{t+n} temporally following time t , all afflicted single-step entropy based rewards $-H(s_{t+n})$ and the κ_1 those steps were performed with earlier, will now be linked to and stored with their $Q(s', \mathbf{a}')$. Those elements L' at n time steps after following \mathbf{a}'_t in s'_t are then accessible via

$$L'(Q(s'_t, \mathbf{a}'_t), t+n) = L'_{t+n} \quad \text{with } L' \in \{\mathbf{a}, -H(s), \kappa_1\}. \quad (13)$$

Please note that regarding the convergence criterion of Reinforcement Learning, episodes' steps are performed randomly ($\epsilon = 0$) at the beginning of the training phase and get an increasingly higher probability ϵ of being performed greedily by applying (7) when training lasts. If now a time step is only followed by random actions, we can find out the stored, but actually not agent-influencing history information L'_{t+n} and replace it with the currently valid parameters, e.g. the current κ_1 . Thus, random episodes, which mainly appear at the beginning of a training phase, can be used as a common database for all κ_1 to be tested. Of course, even greedy steps or whole episodes of an earlier evaluated κ_{1x} can serve as such a random base for each κ_{1y} .

4.2 Demands on Parallel Learning

Having a commonly usable base of random actions and ratings this way, the actual search for κ_1 can begin. To do so, an optimized search within a theoretically unlimited, but actually generously chosen, range of κ_1 is performed via an Adaptive Random Search algorithm [9]. But an enormous problem occurring here again refers to the interdependencies in κ - and Reinforcement Learning since the latter needs to have greedy steps as well as random steps within one episode. On the other hand κ -learning would prefer to rate only episodes performed completely greedily in order to achieve the true CBR. Since the demand is to parallelize the learning algorithms, a compromise has to be found. So what is done is that each first step in an episode is performed greedily, while each following step might be either greedy or random. This proceeding turns out to be acceptable, because first steps should be evaluated well enough within the preceding, all-random training phase, resulting in no drawback for the Reinforcement Learning.

If then an episode's i th step is eventually random we apply an originally Reinforcement Learning sub-strategy to the κ -learning, namely the Q-Learning [10], which is called an off-policy strategy but has been shown to converge to the results of the common on-policy calculation. The method is to just replace the rewards and costs that would have been gained if finishing the episode greedily, by those that are stored in the one $Q(s', \mathbf{a}')$ providing the most influential component when calculating $\widehat{Q}(s, \mathbf{a})$, if continuing greedily. This $Q(s', \mathbf{a}')$ will be called $\widetilde{Q}(s', \mathbf{a}')$ from now on. Since each $Q(s', \mathbf{a}')$ stores an array of experienced rewards and actions over time (see section 4.1), these rewards can now be accessed with the help of (13), leading to (14, 15) where the approximative calculations of the components in (12) are shown. Doing so forces the algorithm to calculate one extra action finding optimization step even though proceeding in a random way actually, but assures we do not lose the whole episode, i.e. not having

performed all the episode’s eventually preceding optimization iterations for nothing and also gaining an at least approximative CBR.

$$\sum_{t=p}^{T-1} \Delta(-H(s_{t+1})) = \sum_{t=p}^{T-1} (\Delta(-H(s))) (\tilde{Q}(s'_p, \pi(s'_p), t + 1)) ; \mathbf{a}_p \in \text{random} \quad (14)$$

$$\sum_{t=p}^{T-1} \frac{C(\mathbf{a}_t)}{\kappa_1} = \sum_{t=p}^{T-1} \frac{C(\mathbf{a})}{\kappa_1} (\tilde{Q}(s'_p, \pi(s'_p), t)) ; \mathbf{a}_p \in \text{random} \quad (15)$$

Obviously a rating for κ_1 becomes more reliable when step number p becomes higher since the approximation is over less steps, but it also depends on the quality of the Q -base growing continuously within a training phase.

4.3 Providing Comparability

Given the Q -base learning supporting feature of complete randomness in choosing each episode’s starting position, this constitutes another problem when trying to learn κ_1 simultaneously. So because of having to compare CBRs of episodes with generally different starting points, κ_{1x} might be disadvantaged to κ_{1y} right from the start if the initial position is worse in a global manner, i.e. if reaching best view points at a greater distance on average. Thus an objectively better κ_1 might be rated worse just because of the awkward initial position of its associated episode. Since we need to provide comparability, the character of an initial position has to influence κ -learning as well, leading to the definition of a state quality $Z(s_t)$ (16) and its integration to the calculation of CBR (17).

$$Z(s_t) = \frac{\int_{\mathbf{a}_t} \psi(s_t) d\mathbf{a}_t}{\max_{\mathbf{a}_t} \psi(s_t)} ; \psi(s_t) = \frac{\sum_{t=p}^{T-1} (\Delta(-H(s))) (\tilde{Q}(s'_p, \mathbf{a}'_p), t + 1)}{\sum_{t=p}^{T-1} \frac{C(\mathbf{a})}{\kappa_1} (\tilde{Q}(s'_p, \mathbf{a}'_p), t)} \quad (16)$$

$$\text{CBR}_{\text{new}} = \text{CBR} \cdot Z(s_0)^{-1} \quad (17)$$

By integrating over the expected long-run rating of all possible immediate actions \mathbf{a}_t in current state s_t in (16) one calculates a state dependent rating, while the normalization in the denominator of $Z(s_t)$ assures we are able to compare even κ_1 learned with different object classes or between episodes occurring at larger distances in the training phase. Integration into the CBR then has to be done inversely since each κ_1 rated by an episode with a convenient starting position should be devaluated adequately and vice versa. It is worth annotating that there is no need or benefit in applying $Z(s_t)$ to any other states $s_{t>0}$ reached within an episode as those are not acquired randomly but are already part of the strategy followed with the actual κ_1 .

Recalling section 2, fortunately most of the above mentioned work has already been done. This is because during the ARS-simplex optimization process (7) for finding the next best action concerning the Q -base, a huge amount of actions has to be evaluated anyway. Noticing that there is a global optimization first in the ARS-simplex, we can just take these global actions’ results for calculating $Z(s_0)$ since the range of actions is sufficiently and almost equally covered, approximately representing the prior integral.

5 Experimental Results

For showing the resulting effect of cost integration, a comprehensible viewpoint selection task was chosen, having to differentiate four kinds of similar cups with either letter A or B on the one side and numeral 1 or 2 at the opposing point (fig.2). The agent is just able to move on a circular line around the cup while each view within the range of ± 60 degrees from a character's center (90° resp. 270°) could be reasonably taken for classification, but with decreasing reliability. When applying the introduced algorithms to this problem there is just one cost-factor ratio $\frac{\kappa_2}{\kappa_1}$ to be considered, where κ_1 is set to be the cost of moving the agent by one degree and κ_2 represents the cost of taking another view. The Reinforcement Learning technique was chosen to be Monte Carlo Learning with $\gamma = 0.2$ and a maximal episode length of five steps.

Learning was then done with 800 training episodes for each cost-factor relation in tab.1, whereas the initial 200 episodes were performed exclusively at random for acquiring a reliable database. During the following 600 episodes the probability of greediness was chosen to increase in a linear way up to $\epsilon = 1$. Taking different ratios m_2^{-1} , tab.1 shows the actually learned κ_1 , the averaged resulting classification rate, the resulting averaged amount of additionally taken views and the averaged performed move length within an episode. Concerning the classification rate, it has to be noticed that this environment was chosen to be very accommodating for keeping the side-effects small in order to bring out the decisive results more clearly. A pre-limitation of $\kappa_1 \in [0.0001; 0.1]$ was arranged.

Regarding tab. 1, it becomes obvious that the system is able to change its behavior decisively when applying costs to the agent's actions. The results show that for this special environment we enabled the agent to save up to more than 50 degrees ($\sim 25\%$) of its rotational movement, compared to a former cost-insensitive learning, without reducing the classification rate. This benefit is due to the fact that the agent now does not reach for the absolute best view on each side of the cup any more, but finds its classification on a less reliable, but obviously still satisfying viewpoint if a gain in cost reduction is available this way. Since doing so is always afflicted with the risk of taking a needless view, this earning increasingly declines when taking views becomes more expensive.

Furthermore we even got the chance of influencing the system by small m_2 in a way that mostly prevents correct classification if this becomes too expensive, which also appears to be intuitive and meaningful. And without losing any prior abilities we can still simulate a cost-insensitive system by just taking extremely large values for m_2 , as the first lines of tab.1 prove. This is due to reserving a cost component for taking views in combination with initiating a multi-step approach.

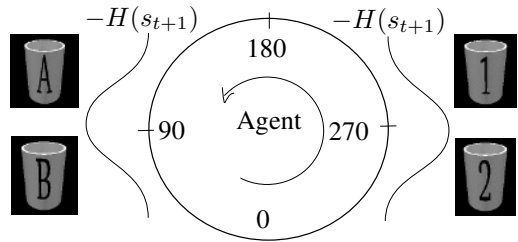


Fig. 2. Arrangement of the evaluated viewpoint selection task

Table 1. Resulting behavior for different cost-factor ratios

m_2^{-1}	learned $\kappa_1 * 10^4$	classification rate [%]	views/episode	move[°]/episode
without costs	-	100	2.47	197.12
1:100000	692.83	100	2.47	198.02
1:10000	937.29	100	2.47	197.33
1:1000	350.23	100	2.48	195.49
1:100	17.23	100	2.51	190.80
1:80	3.30	100	2.55	181.37
1:60	9.46	100	2.59	165.67
1:50	4.71	100	2.66	141.88
1:40	24.13	89	3.02	112.82
1:30	24.76	41	5.17	30.55
1:10	28.39	39	5.60	7.96

6 Summary and Future Work

In this paper it was shown how to expand an already existing Reinforcement Learning based viewpoint selection problem to a more foresighted version. Since the task of object classification could already be solved in an optimal way by taking a minimal amount of views, attention was towards also considering any kind of costs appearing at the same time, e.g. the agent's movement. Integration of those costs to the learning process was revealed, where costs need to be settled in an appropriate range in dependency of the environment's entropy based reward. To do so, a rating function was introduced for learning an appropriate single cost-factor the problem could be reduced to.

Therefore, the main attention was on not increasing the effort of pure, cost-insensitive learning, e.g. the amount of episodes, in a significant manner. So parallel strategies of the basic Reinforcement Learning and the newly introduced cost-factor learning were developed, regarding the reusability of already obtained values as well as coordinating the various differing demands of both techniques at steps within a training phase.

Upcoming work should now tend towards approaches for rating arbitrary cost-factors even by episodes performed with completely different cost-factors, and for eventually making use of completely random episodes for rating. The ability to provide a rating as often as possible is essential, since tasks with broadly higher dimensional action spaces call for a more founded and more reliable evaluation of the cost-factor in order to still find the optimum, while furthermore the cost-factor's reasonable range should become barely pre-limitable beforehand, leading to a larger search space.

References

1. Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, Massachusetts, 1995. Volumes 1 and 2.
2. F. Deinzer, J. Denzler, and H. Niemann. On Fusion of Multiple Views for Active Object Recognition. In B. Radig, editor, *Mustererkennung 2001*, pages 239–245, Heidelberg, September 2001. Springer.

3. F. Deinzer, J. Denzler, and H. Niemann. Viewpoint Selection – Planning Optimal Sequences of Views for Object Recognition. In N. Petkov and M. Westenberg, editors, *Computer Analysis of Images and Patterns – CAIP 2003*, LNCS 2756, pages 65–73, Heidelberg, August 2003. Springer.
4. C. Elkan. Cost-Sensitive Learning and Decision Making When Costs Are Unknown. In *Proceedings of the 17th International Conference on Machine Learning*, 2000.
5. K.S. Hong, K. Ikeuchi, and K.D. Gremban. Minimum Cost Aspect Classification: A Module of a Vision Algorithm Compiler. In *10th International Conference on Pattern Recognition*, pages 65–69, Atlantic City, USA, June 1990.
6. K. Klein and V. Sequeira. View Planning for the 3D Modelling of Real World Scenes. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 943–948, Takamatsu, Japan, 2000.
7. L. Paletta and A. Pinz. Active Object Recognition by View Integration and Reinforcement Learning. In *Robotics and Autonomous Systems*, vol. 31, pages 71–86, 2000.
8. R.S. Sutton and A.G. Barto. *Reinforcement Learning*. A Bradford Book, Cambridge, London, 1998.
9. A. Törn and A. Žilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer, Heidelberg, 1987.
10. C. Watkins and P. Dayan. Q-learning. In *Machine Learning*, 8(3/4), pages 279–292, 1992.

Support Vector Machine Experiments for Road Recognition in High Resolution Images

J.Y. Lai¹, A. Sowmya¹, and J. Trinder²

¹ School of Computer Science and Engineering
{jlai, sowmya}@cse.unsw.edu.au

² School of Surveying and Spatial Information Systems,
University of New South Wales, Sydney, NSW 2052, Australia
j.trinder@unsw.edu.au

Abstract. Support Vector Machines have received considerable attention from the pattern recognition community in recent years. They have been applied to various classical recognition problems achieving comparable or even superior results to classifiers such as neural networks. We investigate the application of Support Vector Machines (SVMs) to the problem of road recognition from remotely sensed images using edge-based features. We present very encouraging results from our experiments, which are comparable to decision tree and neural network classifiers.

1 Introduction

Road extraction from remotely sensed images is an important process in the acquisition and updating of Geographical Information Systems. Automatic and semi-automatic road recognition is an active area of research [7]. RAIL is a road recognition system that has been under development by our group for a number of years. It serves as a framework to research new directions in applying machine learning to image understanding [4, 10].

Support Vector Machines (SVMs) provide a relatively new classification technique that has grown from the field of statistical learning theory [11]. SVMs construct a hyper plane in the feature space that separates the positive and negative training samples. SVMs have been applied to many classic pattern recognition problems with great success including face recognition, hand-written character recognition and speech recognition [1]. In the domain of remote sensing, SVMs have been applied mostly to land cover classification. Camps-Valls et al. [2] use hyper spectral data of 128 bands to classify 6 types of crops. SVM yielded better outcome than neural networks. SVMs also performed reasonably well in situations where feature selection was not used. Pal and Mather [8] report that SVMs performed better than maximum likelihood, univariate decision tree and back-propagation neural network classifier, even with small training data sets. Both groups used pixel-based features.

There are two main motivations to incorporate SVMs into RAIL. First of all, SVMs have been successful in other application domains. However, there have been no re-

sults (prior to [13]) published on applying SVMs to the problem of road recognition. Therefore, our experiment will be of interest to pattern recognition communities as well as remote sensing researchers. Secondly, RAIL uses a meta-learning framework that facilitates model selection for classifiers, amongst other types of learning. Incorporating SVMs into RAIL expands the base algorithm sets to promote meta-learning research.

This paper is organised as follows. Section 2 describes implementation improvements on RAIL. Section 3 describes the experiment and the results are presented in Section 4. We summarise our results in Section 5.

2 RAIL

RAIL is an adaptive and trainable multi-level edge-based road extraction system which has been developed within our group for a number of years [10]. Starting with low-level objects (edges), RAIL incrementally builds higher-level objects (road network). The levels of classification are

1. Road Edge Pairs - pairs of edges that enclose a segment of road
2. Linked Road Edge Pairs - adjacent road edge pairs that form continuous roads
3. Intersections - road edge pairs that meet to form intersections
4. Road Network - linked roads and intersections.

SVM was applied to the preprocessing stage (edge extraction) and Level 1 of RAIL with encouraging results [13]. This paper extends the use of SVM to Level 2 while removing SVM use in the preprocessing stage. Several implementation improvements have been made to RAIL that affected the previous SVM experimentation. These include the image processing stage, the reference model, feature extraction and feature selection stages.

Image Processing: The parameters used in Vista's Canny edge-detector were tuned to produce outputs with less noise. This was accomplished by adding noise to the original image prior to a Gaussian smoothing function with a large standard deviation. Adding artificial noise to our images before blurring removes very small features such as noise that are present in high resolution images. The improvement was a dramatic decrease in the number of extracted edges, up to 90% less in several images, which meant that SVM could be used to learn Level 1 data without an additional SVM preprocessing. Removing this preprocessing stage gives results that can be compared to other algorithms in RAIL which also do not use any additional preprocessing stage. Another advantage is the reduction in misclassification during the SVM preprocessing stage (approximately 14%) so that a more complete road network can be recovered at higher levels.

Reference Model: RAIL has recently adopted a centreline reference model based on Wiedemann et al. [12] which can assess the learned outputs more correctly by checking

Table 1. Extracted Features

Level 1	Level 2
Width (mean)	Width (mean)
Enclosed Intensity (mean)	Width (var)
Enclosed Intensity (var)	Width Difference
Pair Length (centreline)	Enclosed Intensity (mean)
Length Difference	Enclosed Intensity (var)
Bearing Difference	Enclosed Intensity Difference
Intensity Gradient Difference	Gap Intensity (mean)
Projection	Gap Intensity (var)
	Length Combined
	Length Difference
	Minimum Gap Separation
	Maximum Gap Separation
	Gap Separation (mean)
	Bearing Difference
	Intensity Gradient Difference (left)
	Intensity Gradient Difference (right)

that the extracted pairs have edges that do in fact lie opposite each other near the reference model. Previously we used an edge based model which produced a slightly more modest value in assessing the correctness of the outputs.

Feature Extraction: Additional features have been added to Level 1 and Level 2 (see Table 1) and a relevant subset from each level was selected by using feature subset selection (FSS) methods, which is described in section 3.4. The highlighted entries are the feature subsets that were discovered. Descriptions of the features may be found in [6].

Selected Level 1 Features: Pair width, enclosed intensity (mean), bearing and projection form an intuitive feature subset that describes road segments, i.e. roads have similar width and intensity and their opposite sides are almost parallel. Pair length is a good feature because in our preprocessing stage we have set a maximum length for edges. Generally road sides are long and continuous and get split into smaller segments after preprocessing. When road pairs are formed their lengths do not vary too much. This is because non-road edges are usually of shorter length.

Enclosed intensity variance did not prove to be a good feature since the area enclosed by an edge pair is small and the intensity is fairly similar. Length difference between edges was also discarded by FSS. We expect road pairs to have similar edge length but non-road pairs maybe also have similar edge lengths, thus it does not convey much information. Intensity gradient difference between the two edges do not show consistencies between road pairs and non-road pairs. The assumption that the intensity levels are the same on both the external sides of the road is invalid.

Selected Level 2 Features: At Level 2, linked road pairs should have similar enclosed intensity with little difference. Ideally linked pairs should be minimally separated and

have no gap, thus *gap intensity* and *gap separation* are excellent features to distinguish between linked road pairs and other linked edge pairs. Roads generally have smooth curves except at an intersection, therefore the *bearing difference* between linked road pairs should not be very large.

Width features are not good attributes for Level 2 because Level 1 outputs all have similar widths. The same argument applies to length attributes. *Enclosed intensity variance* and *gap intensity variance* are not very good features for the same reason discussed earlier, i.e. intensity level do not change much in enclosed edge pair or in a road gap. Again, intensity levels across edges cannot be assumed to be the same on both sides of the linked edge pairs.

Feature Subset Selection: The goal of FSS is to choose the most relevant features for classification, in other words, removing irrelevant attributes that may distract a machine learning algorithm. We compiled 9 sets of data from our images. 7 were from individual images and 2 were random selections from all the images. The sample size ranges from 130 to 360 examples in each set. We did not use one large test set since we had different road types and having one set of data might cause the result to be biased towards the most frequent road type.

The Weka data mining suite (version 3.4)¹ was used to conduct the FSS experiments. The FSS algorithms are grouped into *Type I*, consisting of correlation-based, classifier and wrapper algorithms, and *Type 2*, consisting of Chi squared, Relief, Information Gain, Gain Ratio and Symmetrical uncertainty algorithms. Type I algorithms select the ‘best’ subset of features. The frequency of each attribute was recorded and averaged. Type II algorithms rank the individual attributes by assigning them a weighting. These were normalised and averaged.

We wanted to select a subset of features which have a high frequency score in Type I and a high weighting in Type II. We ranked the Type I and Type II results and picked the smallest subset where the features are the same in each type. For example, if the top 4 attributes in Type I and Type II are the same disregarding their relative ranking position, then we would have a subset of 4 features. This has produced good classification results.

Features Versus Heuristic Preprocessing: Although we are using new image processing parameters to produce less noisy outputs, we are still dealing with fairly large datasets for Level 1 (C_2^n), as each edge can be paired up with every other edge and the ordering is irrelevant). Thus we use heuristic preprocessing to reduce the data size so that it becomes more manageable. We do not use heuristic rules for Level 2 since the data size is comparatively smaller than Level 1.

The heuristic rules throw away cases where an expert would agree that a positive classification is impossible. For example, in Level 1 we used the heuristic that if edges in an edge pair do not project onto each other, then they cannot be classified as an edge pair, since they are not opposite each other. Because this feature has a binary output, by using this attribute as a heuristic filter we have effectively removed projection from the feature space, since the heuristic rule outputs only those edge pairs that do project

¹ Software available at <http://www.cs.waikato.ac.nz/ml/weka>

on to each other. We also have a heuristic rule that leaves out any edge pairs that are wider than twice the maximum road width in the images. We have effectively reduced the feature space that SVM would need to learn from.

Theoretically this should not make any difference to machine learning algorithms because the data we are leaving out have no influence on how the classes are separated. For SVMs, the data points discarded are distant from the class separation region and the support vectors, thus the construction of the separation hyper plane is independent of them.

Dataset: Seven high resolution aerial images were used in the experiment. Image A and B are from a rural area in France. These images have a ground resolution of 0.45m/pixel. The other 5 images are from a rural area in Morpeth in Australia. These images have a ground resolution of 0.6m/pixel. The image properties are given in Table 2.

Table 2. Image Properties

Image	Dimensions	No. of Edges
A	776*289	1530
B	757*821	3055
C	1500*848	2912
D	1700*1300	3290
E	1400*1300	1858
F	1400*1200	3893
G	1600*1100	3204

A total of 333 and 227 positive and negative examples were selected from the images (some images contain more examples) for Level 1 and Level 2 respectively. The heuristic preprocessing outputs serve as inputs data for Level 1, and the Level 1 outputs feed into Level 2. The size of the test data ranges from 2400 to 11200 instances for Level 1 and between 1500 to 18200 instances for Level 2.

Since we only had seven images to experiment with, we used 7-fold cross validation technique (leave-one-out) for evaluating the learned output, i.e. we train using six images and test on the unseen image. Note however that at the edge pair and twin linked edge pair level where the learning takes place, we have thousands of instances in each image.

3 Experimental Design

SVM experiments have been conducted on Level 1 and Level 2 of RAIL (the level references are different to those in [13]). The SVM implementation used was changed to LIBSVM² (version 2.4) which offers more in terms of tools and programming interfaces.

² Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

The training data and test data were scaled to $[-1,1]$ to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during SVM calculations [3].

We used five different kernels for training SVMs for Support Vector Classification (C-SVC). They can be separated into two categories: Polynomial and Radial Basis Function (RBF). The polynomial kernels are of the first, second and third degree (with default $C=1$). The RBF kernels are standard (RBFs, $C=1$, $\gamma=1$) and optimised (RBFo, C , γ picked by a grid search function provided by LIBSVM). C is the penalty parameter that controls the margin and hence the over fitting of data, and γ is an internal variable for RBF.

The SVM kernels are compared to two well known classifiers within Weka, namely decision tree (DT) and neural network (NN), with default settings.

4 Experimental Results

The metrics used to evaluate the results are taken from Wiedemann et al. [12]. They address two questions: 1. How complete is the extracted road network, and 2. How correct is the classification. They are calculated to percentage values, given by:

$$completeness = \frac{length_{TP}}{length_{reference}} \quad (1)$$

$$correctness = \frac{length_{TP}}{length_{classified}} \quad (2)$$

Completeness measures the percentage of the road reference as detected by SVM. Correctness measures the percentage of the SVM classification that are actual road pairs. A high completeness means that SVM has extracted most of the road network, whereas high correctness implies that SVM has not classified too many incorrect road pairs.

We combine the two measures above into a more general measure of the quality. We call this *cxc* which is expressed as:

$$cxc = completeness^2 * correctness \quad (3)$$

Clearly, this measure is biased towards completeness. RAIL uses the output of Level 1 as the input of Level 2, so it is more important to have high completeness at the lower levels for input to higher levels. For example, Level 2 will only be as complete as its input (Level 1 output). Higher correctness value will result as higher levels discard non-road pairs.

Tables 3 shows the classification results (rounded to nearest percent) for Level 1 and Level 2. The completeness (cmp), correctness (cor) and *cxc* are shown for each classifier on each image. The entry with the highest *cxc* for each image in Level 1 is used as input to Level 2. The highest *cxc* obtained by SVM classifier has been highlighted for each image. Fig. 1 to Fig. 4 shows the SVM results visually for Image F. The images consist of the road reference, the low-level edges as inputs and the best Level 1 and Level 2 outputs.

Some of the completeness values are a little over 100%, this is because the centreline reference model uses a buffer zone both to the left and to the right of the road reference. Although the buffer width is only set to 3 pixels on either side, on some noisy road sections, two or more edges maybe measured as true positives for that same section. However, this is only true in a few cases. In all images with completeness greater than 100%, detailed analysis show that more than 98% of the reference road network is recognised.

Level 1 SVM classifiers have an average of 97% completeness and 35% correctness. Level 2 SVM classifiers have an average of 90% completeness and 49% correctness. These results are very encouraging because high completeness values are obtained. Clearly, the polynomial kernel of degree 3 and the optimized RBF kernel outperform the other kernels except for Image E. Additionally, the SVM classifiers compare well to DT and NN classifiers. In most cases, the results are very similar, though on images



Fig. 1. Image F - Road Reference

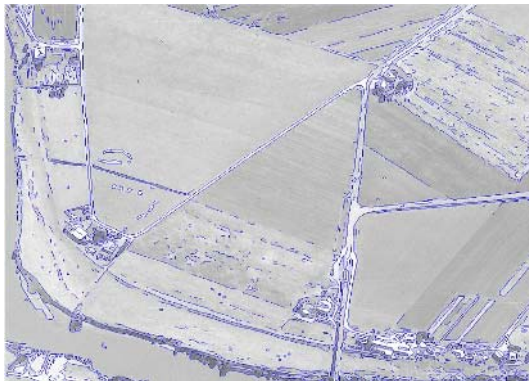


Fig. 2. Image F - Input

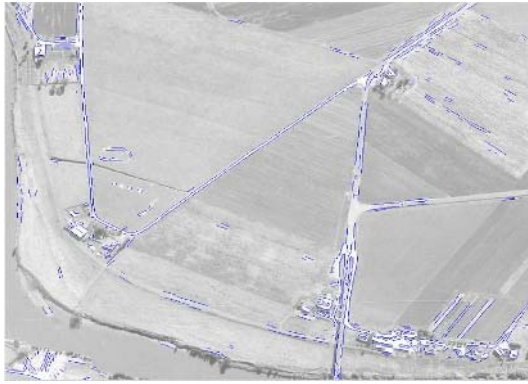


Fig. 3. Image F - Level 1 output



Fig. 4. Image F - Level 2 output

containing dirt roads in Level 1 (Image E and F), SVM classifiers appear to outperform both DT and NN, see Table 3.

The low correctness value in Level 1 does not worry us. One of the major causes of the large number of false positives is that SVM classified road pairs have similar road properties, but only a fraction of them are represented by the centreline road reference. The others appear to be driveways and crop separations (perhaps for tractors) which are non-roads, but picked up well by the classifiers. The other main reason is that road properties may vary slightly between different images. SVMs learn these variations to a certain degree and thus the classified output may contain a range of road properties, some of which might be non-roads depending on the images.

Some images had lower completeness in Level 2, particularly Images C, E and F. The main causes of this are 1) because the road is very similar to its surroundings (especially roads with lower intensity), which means edges are not extracted well, and 2) dirt roads have been misclassified in Level 2 since the edge pairs are not closely

Table 3. Classification Results

	Classifier	L1 Cmp	L1 Cor	L1 cxc	L2 Cmp	L2 Cor	L2 cxc
A	Poly. 1	101	32	34	98	53	50
A	Poly. 2	100	32	33	98	54	51
A	Poly. 3	101	37	38	98	53	51
A	RBFs	101	35	36	98	54	51
A	RBFo	101	35	36	98	54	52
A	DT	100	31	31	98	54	51
A	NN	101	36	37	98	54	51
B	Poly. 1	107	34	39	105	51	57
B	Poly. 2	96	18	17	105	51	58
B	Poly. 3	94	39	34	105	54	60
B	RBFs	103	36	38	105	52	57
B	RBFo	108	36	42	105	53	59
B	DT	102	29	30	105	52	57
B	NN	107	39	44	105	51	56
C	Poly. 1	92	26	23	81	41	27
C	Poly. 2	89	21	17	81	41	27
C	Poly. 3	87	33	25	81	41	27
C	RBFs	94	29	25	81	41	27
C	RBFo	94	30	27	81	42	27
C	DT	92	29	25	81	41	27
C	NN	92	31	26	81	41	27
D	Poly. 1	100	22	21	98	30	28
D	Poly. 2	98	23	23	98	30	29
D	Poly. 3	97	23	22	98	30	29
D	RBFs	99	23	22	98	30	28
D	RBFo	99	24	24	98	30	29
D	DT	97	27	26	98	30	29
D	NN	98	26	25	98	30	28
E	Poly. 1	85	47	34	84	70	49
E	Poly. 2	80	43	28	84	69	48
E	Poly. 3	81	56	37	84	66	46
E	RBFs	91	56	46	84	70	49
E	RBFo	87	57	43	84	70	49
E	DT	37	41	6	84	70	49
E	NN	51	55	15	84	70	49
F	Poly. 1	83	32	22	68	54	25
F	Poly. 2	91	27	22	68	55	25
F	Poly. 3	88	37	29	68	55	25
F	RBFs	88	35	27	68	55	25
F	RBFo	84	33	24	67	55	25
F	DT	70	36	18	67	55	25
F	NN	73	43	23	68	54	25
G	Poly. 1	98	35	34	99	42	40
G	Poly. 2	97	30	28	98	43	40
G	Poly. 3	96	39	36	99	43	41
G	RBFs	100	38	38	99	42	40
G	RBFo	100	38	37	99	42	41
G	DT	92	31	27	96	42	39
G	NN	95	34	30	99	42	41

linked. Fig. 4 is a good example where narrower roads with high intensity have been detected while wider and lower intensity roads have been missed. This problem can be fixed by applying a further preprocessing stage before edge extraction, e.g. multi-level thresholding/segmentation or by using an ensemble of SVMs and combining the results.

5 Conclusions

In this paper we have experimented with SVM and road extraction using edge-based features, which is significantly different from other SVM experiments in the remote sensing domain. The results for Level 1 and Level 2 are very encouraging and comparable to decision trees and neural networks. We plan to extend SVM to level 3 of RAIL which currently uses a relational learning algorithm to recognise the attributes of junctions [9].

The current experiments also show that it is feasible to select a suitable kernel that is best for the data. In the future we plan to experiment with other kernel functions and apply meta-learning techniques to find the best kernel and the parameters that are associated with them (Gold and Sollich, 2003).

References

1. Byun, H., Lee, S., 2003. A survey on pattern recognition applications of support vector machines. *IJPRAI*, 17(3), pp. 459-486.
2. Camps-Valls, G., Gomez-Chova, L., Calpe-Maravilla, J. Soria-Olivas, E., Martin-Guerrero, J. D., Moreno, J., 2003. Support Vector Machines for Crop Classification Using Hyper spectral Data. *Proc. IBPRIA 2003*, pp. 134-141.
3. Chang, C., Lin, C., 2001. LIBSVM: A library for support vector machines. Available at <http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf> (accessed 28 Feb. 2004).
4. Chen, A., Donovan, G., Sowmya, A., Trinder, J., 2002. Inductive Clustering: Automating low level segmentation in high resolution images. *Proc. ISPRS Commission III Symp. Photogrammetric Computer Vision*, pp. 73-78.
5. Gold, C., Sollich, P., 2003. Model selection for support vector machine classification. *Neurocomputing*, pp. 221-249.
6. Lai, J., Sowmya, A. and Trinder, J., 2004. Support Vector Machine Experiments for Road Recognition in High Resolution Images, Tech. Report, School of Comp. Sci. and Engg, University of New South Wales, UNSW CSE TR 0413.
7. Mena, J. B., 2003. State of the art on automatic road extraction for GIS update: A novel classification. *Pattern Recognition Letters*, 24(16), pp. 3037-3058.
8. Pal, M., Mather, P. M., 2004. Assessment of the effectiveness of support vector machines for hyper spectral data. *Future Generation Computer Systems*, Article in Press.
9. Nguyen, H. S., Sowmya, A., Trinder, J., 2005. Improved Road Junction Recognition in High Resolution Images using Relational Learning. (personal communication).
10. Sowmya, A., Singh, S., 1999. RAIL: Extracting road segments from aerial images using machine learning. *Proc. ICML 99 Workshop on Learning in Vision*, pp. 8-19.
11. Vapnik, V., 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

12. Wiedemann, C., Heipke, C., Mayer, H., Jamet, O., 1998. Empirical Evaluation of Automatically Extracted Road Axes. In *CVPR Workshop on Empirical Evaluation Methods in Computer Vision*, pp. 172-187.
13. Yager, N., Sowmya, A., 2003. Support Vector Machines for Road Extraction from Remotely Sensed Images. *Proc. CAIP 2003*, pp. 285-292.

An Automatic Face Recognition System in the Near Infrared Spectrum

Shuyan Zhao and Rolf-Rainer Grigat

Technical University Hamburg Harburg, Vision Systems,
4-08/1, Harburger Schloßstr 20, 21079 Hamburg, Germany
{s.zhao, grigat}@tu-harburg.de

Abstract. Face recognition is a challenging visual classification task, especially when the lighting conditions can not be controlled. In this paper, we present an automatic face recognition system in the near infrared (IR) spectrum instead of the visible band. By making use of the near infrared band, it is possible for the system to work under very dark visual illumination conditions. A simple hardware enables efficient eye localization, thus the face can be easily detected based on the position of the eyes. This system exploits the feature extraction capabilities of the Discrete Cosine Transform (DCT) which can be calculated very fast. Support Vector Machines (SVMs) are used for classification. The effectiveness of our system is verified by experimental results.

1 Introduction

Face recognition has a wide variety of applications in commercial and law enforcement. Due to its nonintrusive characteristics, it emerges as a vivid research field in biometrics. Illumination is one of the challenges for face recognition. Most face recognition algorithms are associated with visible spectrum imagery, thus they are subject to changeable lighting conditions. For systems that have to work in the daytime and at night, infrared is a solution. However, thermal infrared is not desirable because of the higher cost of thermal sensors and poorer quality of the thermal images. Therefore near infrared is preferable and common silicon sensors can be used, since they are sensitive from the visible band to near infrared band (up to 1100 nm).

An automatic face recognition system must detect the face first. It requires either face detection as the first module, or localization eyes without face detection and cropping the face region accordingly. Since alignment is very significant for face recognition, it is advantageous to detect eyes first. When the eyes are localized, the face region can be segmented and aligned in accordance with the reference faces, in which case good recognition performance can be expected. On the other hand, precise face detection is a very tough task, so if recognition relies on the uncorrect face region, the system performance will be degraded. Moreover, when the lighting source is placed close to the camera axis oriented toward the face, the interior of the eyes reflects the light and pupils appear very bright.

This is the well-known “bright pupil” (red-eye) effect [1] and can be exploited to detect eyes.

Chellappa [2] and Zhao et al. [3] presented nice surveys of face recognition algorithms. Face recognition technology falls into three main categories: feature-based methods, holistic methods as well as hybrid methods. Feature-based approaches depend on the individual facial features, such as the eyes, nose and mouth, and the geometrical relationships among them. A representative of feature-based methods is *Elastic Bunch Graph Matching* [4]. Holistic methods take the entire face into account. Among global algorithms, the appearance-based methods are the most popular, for example, *Eigenface* [5], *Fisherface* [6], and *Independent Component Analysis* (ICA) [7]. Hybrid methods combine the feature-based and holistic methods, for instance, the algorithm presented in [8] is a combination of *Eigenface* and *Eigenmodule*. Recently, Huang et al. [9] proposed a hybrid method, which incorporated the component-based recognition with 3D morphable models.

In this paper, we propose a face recognition system for access control in the near infrared spectrum. A simple and low cost hardware has been built up and used for data collection, which is presented in the next section. In Section 3, the algorithms for automatic face recognition are introduced. “Bright pupil” effect is utilized to localize the eyes, based on which the face is detected. DCT coefficients are then selected as features, Support Vector Machines (SVM) are employed to identify faces. Experimental results are shown in Section 4. In the last section conclusions are drawn and an outlook is given.

2 Hardware Configuration and Data Collection

2.1 Hardware

In order to make use of the “bright pupil” effect, a lighting source along the camera axis is necessary. In the literature [10] [1], the camera was equipped with two lighting sources, one along the camera axis and the other far from the camera axis. When the on-axis illuminator is switched on, the bright pupil image is generated; when the off-axis illuminator is on, the dark pupil image is produced. The difference of the two images gives clues about the position of the eyes. However, such a system needs a switch control which has to be synchronized with the frame rate so that the even and odd frames correspond to the bright and dark images respectively.

In our system a simplified hardware is used. An illuminating ring, consisting of 12 IR-LEDs, is placed along the axis of an inexpensive CCD camera. Thus only bright images can be generated by our system. The dark image will be constructed by using software rather than hardware. In order to obtain stable illumination conditions, an IR filter is used to block the visible light. The hardware is shown in Fig. 1.

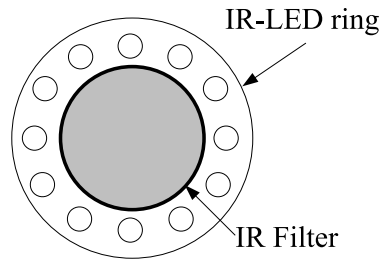


Fig. 1. The IR-LED ring and the IR filter



Fig. 2. Some examples of test data



Fig. 3. Some examples of training data

2.2 Data Collection

There are a couple of available face databases, but as far as we know, all of them contain face images taken under day light conditions. Therefore it is necessary for us to collect data ourselves in order to develop algorithms for face recognition in the near IR spectrum.

Reference data and test data were collected respectively. Our motivation is to realize access control in the situation where a camera is set above the door under protection. Therefore video sequences were captured under such a condition as test data, examples can be found in Fig. 2. Still images of frontal faces were taken as reference data, see Fig. 3.

In order to include variations of facial expressions, the subjects were requested to speak vowels a, e, i, o, and u. The resolution of all the images is 320×240 . Images of 10 subjects from Europe, Asia and Africa have been collected so far.

3 The Automatic Face Recognition System

In the automatic face recognition system, the first step is eye localization. Based on the eye positions, the face region will be segmented and then normalized

to a standard size. In the normalized face image, two eyes are located at the predefined positions. DCT features are then extracted, finally SVM is employed to recognize the face.

3.1 Eye Localization

The algorithm for eye localization is shown in Fig. 4. The morphological operator, opening, is at first applied to the original image. This operation removes the bright pupils. Then a difference image is obtained by subtracting the dark image from the original (bright) image, which contains only the bright areas. This difference image is then binarized. The connected components in the binary image are considered as pupil candidates that will be further verified.

The iris of the eye has a circular boundary, which can be detected due to the contrast between the iris and the area around it. Edges of the original image are detected using Canny’s algorithm [11]. In the edge image, a window surrounding each pupil candidate is chosen, then Hough transform [11] is exploited to detect circles inside the window. Those candidates without a circular boundary are noise. Finally pair check is performed, and the two eyes are localized. The stepwise results are illustrated in Fig. 5.

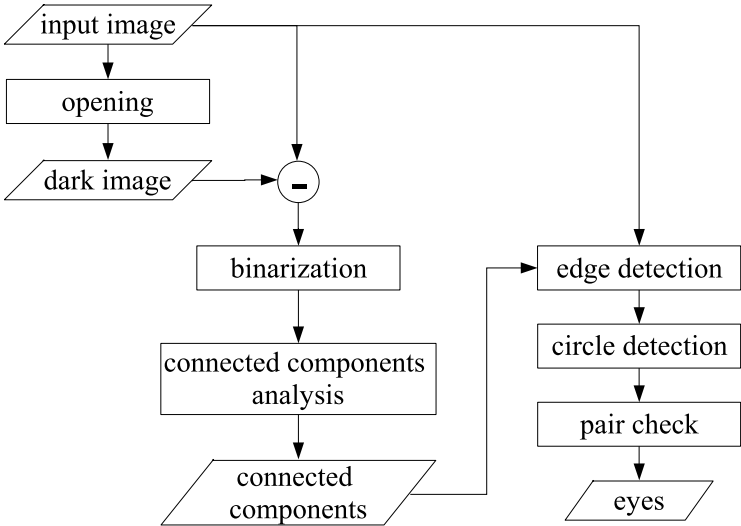


Fig. 4. The procedure of eye localization

3.2 Face Recognition

Discrete Cosine Transform shares a closely related mathematical background with Eigenfaces. However, its merits over Eigenface are: it needs less training

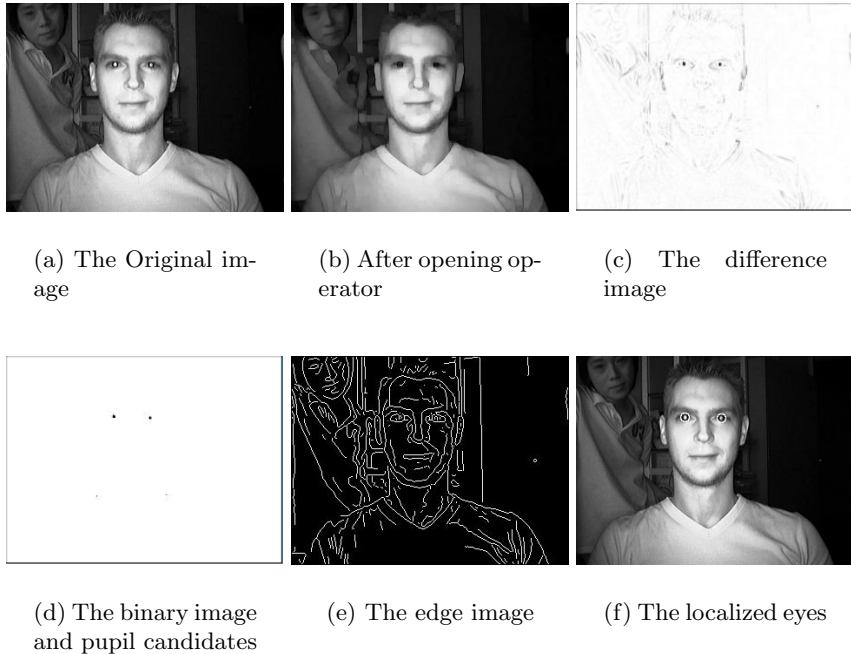


Fig. 5. Stepwise results of eye localization

time; it is deterministic and does not require the specification of the data set. It seems more desirable when new faces are added into the database frequently. Besides, fast algorithms are available to calculate DCT.

2-dimensional DCT is performed on the normalized face. Only a subset of the DCT coefficients at the lowest frequencies is selected as a feature vector. These DCT coefficients have the highest variance and are sufficient to represent a face. To further reduce variations of illumination, these DCT coefficients are normalized to the DC component. SVMs [12] are used as the classifier.

4 Experiments

Three experiments are carried out. The first two experiments test the performance of eye localization and face recognition independently, and the third experiment tests the performance of the complete system.

Eye Localization. 300 images are chosen to evaluate the eye localization performance. The criterion for correct localization is that any estimated eye position within a circle of radius 3 pixels from the true position is counted as a correct detection. In our test, 87.7% (263/300) accuracy is obtained, some results are shown in Fig. 6. The algorithm works when the horizontal distance of two eyes



Fig. 6. Results of eye localization

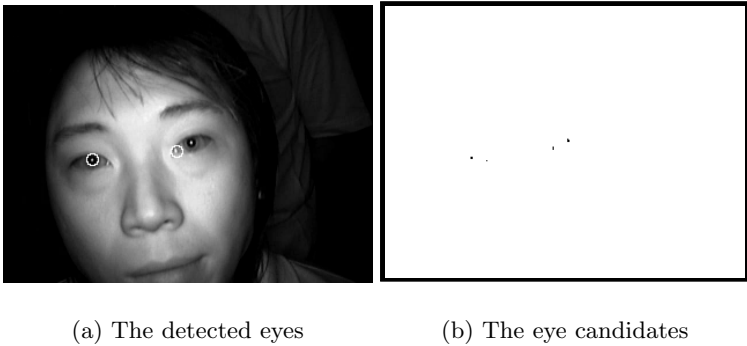


Fig. 7. A false acceptance

is larger than 30 pixels, and the in-plane rotation of the face is less than 25° . If the noise is too close to the true eye, the algorithm may fail, see Fig. 7.

Face Recognition. In this experiment, only the performance of face recognition is taken into account, thus the centers of the eyes are manually marked. 250 face images (25 images per subject) are selected from our reference data set, collected in Section 2, to evaluate the algorithm. These images were captured at different photo sessions so that they display different illumination and facial expressions, even slight pose variations, see Fig. 8. Among them 120 images (12 images per subject) are randomly chosen for training and the left images for testing.

All the faces are scaled to the size 48×48 , aligned according to the eye positions, and histogram equalized. 64 (8×8) DCT coefficients are extracted



Fig. 8. Examples used to test the performance of face recognition

as features. LIBSVM [12], a library for Support Vector Machines, was used to performance recognition, where RBF kernel was employed. Recognition accuracy of 96.15% (125/130) has been achieved.

Automatic Face Recognition. The same face recognition algorithm as the last experiment is used. By integrating with the eye localization module, the automatic face recognition system is tested. 500 images selected from the test data set collected in Section 2 are used for evaluation. These images demonstrate a lot of difference from the training data, because they were captured in different sessions and by a camera mounted above the door, as shown in Fig. 2. Although it is assumed that the subjects are cooperative, pose variations are not impossible. 398 faces are correctly recognized, corresponding to 79.6% accuracy. The reasons for the degraded performance are: 1) The imprecise eye localization results in uncorrect alignment; 2) perspective distortion exits because of the position of the camera, i. e. above the door.

5 Conclusion and Future Work

We present a robust and low cost system for automatic face recognition in the near infrared spectrum. By using the near infrared band, a stable illumination condition is obtained. Face images in the near infrared spectrum have been collected in our laboratory. The system utilizes the "bright pupil" effect to detect eyes. Relying on a series of simple image processing operations, eye localization is efficient. The face region is segmented and aligned according to the position of the eyes. DCT coefficients are selected as features, and the powerful machine learning algorithm Support Vector Machines is used for classification. The experimental results show that good performance is achieved by using our system.

Future Work. Pose variation is the toughest problem for face recognition. In the near future, the functionality of the system will be extended to be able to recognize faces with pose variations.

References

1. Morimoto, C.H., Koons, D., Amir, A., Flickner, M.: Pupil detection and tracking using multiple light sources. *Image and Vision Computing* **18** (2000) 331–335
2. Chellappa, R., Wilson, C., Sirohey, S.: Human and machine recognition of faces: A survey. *IEEE*. **83** (1995) 705–740
3. Zhao, W., Chellappa, R., Rosenfeld, A., Phillips, P.: Face recognition: A literature survey. Technical Report CAR-TR-948, University of Maryland (2000)
4. Wiskott, L., Fellous, J.M., Krüger, N., Malsburg, C.B.: Face recognition by elastic bunch graph matching. In: Proc. 7th Intern. Conf. on Computer Analysis of Images and Patterns, CAIP'97, Kiel. (1997) 456–463
5. Turk, M., Pentland, A.: Eigenfaces for Recognition. *Journal of Cognitive Neuroscience* **3** (1991) 77–86
6. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **19** (1997) 45–58
7. Bartlett, M.S., Lades, H.M., Sejnowski, T.J.: Independent component representations for face recognition. In: Proc. of SPIE Symposium on Electronic Imaging. (1998) 528–539
8. Pentland, A., Moghaddam, B., Starner, T.: View-based and modular eigenspaces for face recognition. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94), Seattle, WA (1994) 84–91
9. Huang, J., Heisele, B., Blanz, V.: Component-based face recognition with 3D morphable models. In: Proc. 4th Audio-Video-Based Biometric Person Authentication, Guildford, UK (2003) 27–34
10. Haro, A., Flickner, M., Essa, I.: Detecting and tracking eyes by using their physiological properties, dynamics, and appearance. In: Proc. of IEEE. Conf. Computer Vision and Pattern Recognition (CVPR2000). (2000) 163–168
11. Pratt, W.K.: *Digital Image Processing*, Third Edition. Wiley-Interscience, New York (2001)
12. Chang, C.C., Lin, C.J.: libsvm (2005) <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.

Hierarchical Partitions for Content Image Retrieval from Large-Scale Database

Dmitry Kinoshenko¹, Vladimir Mashtalir¹, Elena Yegorova²,
and Vladimir Vinarsky³

¹ Kharkov National University of Radio Electronics, computer science faculty,
Lenin Ave., 14, 61166, Kharkov, Ukraine

{Kinoshenko, Mashtalir}@kture.kharkov.ua

² Wessex Institute of Technology,
Ashurst Lodge, Ashurst, Southampton, SO40 7AA, UK
Yegorova@kture.kharkov.ua

³ Loewe Opta GmbH, Kompetenzzentrum Hannover,
Hildesheimer St. 140, D-30173 Hannover, Germany
vvinarski@loewe-komp.de

Abstract. Increasing of multimedia applications in commerce, biometrics, science, entertainments etc. leads to a great need of processing of digital visual content stored in very large databases. Many systems combine visual features and metadata analysis to solve the semantic gap between low-level visual features and high-level human concept, i.e. there arises a great interest in content-based image retrieval (CBIR) systems. As retrieval is computationally expensive, one of the most challenging moments in CBIR is minimizing of the retrieval process time. Widespread clustering techniques allow to group similar images in terms of their features proximity. The number of matches can be greatly reduced, but there is no guarantee that the global optimum solution is obtained. We propose a novel hierarchical clustering of image collections with objective function encompassing goals to number of matches at a search stage. Offered method enables construction of image retrieval systems with minimal query time.

1 Introduction

Short retrieval time independent of the database size is a fundamental requirement of any user friendly content-based image retrieval (CBIR) system. Visual contents of an image such as color, shape, texture, region relations play dominating role in propagation of features selection, indexing, user query and interaction, database management techniques. To search images in a large-scale image database traditionally queries ‘ad exemplum’ are used. Characteristics of different CBIR schemes, similarities or distances between the feature vectors of the query by example or sketch and those of the images collection are sufficiently full explored [1–7]. Essential efforts are devoted to synthesis and analysis of image content descriptors, namely color moments, histograms, coherence vectors, correlograms, invariant color features [8,9]; texture statistical and struc-

tural properties, determining by methods based on Fourier power spectra, Markov random fields, Gabor and wavelet transforms, fractal models, principal component analysis [10–13]; region-based and boundary-based features of shape, salient points in images [14–16]; syntax and semantic representations [17,18]. They form feature spaces with reduced dimensions, however, when very large image collections are in use, matching methods are computationally expensive. One of the widespread approaches to minimize time outlay consists in application of various techniques of image database preliminary processing including clustering methods [7,10,18–20]. Thus a hierarchical mechanism of the query processing can be formed: one can seek suitable clusters in nested partitions with an arbitrary indexing scheme. This way the amount of matches can be greatly reduced, but traditional clustering methods do not guarantee the optimal result.

To optimize this CBIR scheme it is necessary to minimize a total number of matches at the retrieval stage. We propose a new hierarchical clustering which allows to construct images partitions into disjoint subsets so that firstly one can seek suitable class, then the most similar to the query subclass is chosen and so on. The exhaustive search is fulfilled only on the lower level of hierarchy. In the second section formalization of clustering problems is offered. In the third section clustering which guarantees the search for the minimal number of matching is discussed. Thus our contribution consists in development and theoretical ground of novel hierarchical partition construction for the fast content-based image retrieval in video databases.

2 Hierarchical Partitions Formalization for Fast Retrieval

In large databases an efficiency of image retrieval procedures with search 'ad exemplum' is determined in general by two characteristics: by reliability (in terms of precise identification of the required images) and computational complexity (generally, in sense of a matching operations amount) of algorithms used to pick up.

Usually first problem solution is coupled with the choice of features space which is adequate to application area. Image retrieval by example methods require that an index of proximity, or likeness, or affinity, or association has to be established between pairs of any images and such proximity functions can be expressed qualitatively and quantitatively, it can be represented by arbitrary metric or similarity (dissimilarity) measure. Further without loss of generality, let us suppose a metric is sufficient form of proximity function representation and we shall use notation $\rho(o, o)$, where arguments symbolize images in signal or feature space.

At second case, to avoid the combinatorial explosion a large collection of preliminary processing algorithms is available to reduce the search. Often clustering algorithms are used for computing complexity depreciation. Most of them are based on the two popular approaches: agglomerative hierarchical clustering and iterative partitional clustering. Hierarchical techniques organize data in a nested sequence of groups and partition techniques provide obtaining of one-level similar images classes.

Various heuristics are used to reduce a search complexity, but still there is no guarantee of optimal results. As a rule, to guarantee that a global optimum solution has been obtained, one has to examine all possible partitions, which is not computationally feasible. In this connexion let us consider original schemes with the best guaranteed results (with reference to a matches amount) in worst-case conditions.

Let $X = \{x_v\}_{v \in V}$ be a set of features representing images in the database (specifically it is possible to consider signal space also), hereinafter V, Γ, A are index sets. Generally speaking, each feature vector belongs to Cartesian product $x_\alpha \in Z_1 \times Z_2 \times \dots \times Z_r$ of arbitrary nature elements permitting to take into account properties of color distribution, shapes, texture, image semantic etc. Let $Y = \{y_\gamma\}_{\gamma \in \Gamma}$ be a feature set of an arbitrary query image.

The problem is in that under given query $y \in Y$ one needs to find the most similar image (or images) $x_v \in X$. In other words, it is necessary to provide $\min_{v \in V} \rho(y, x_v)$ during minimum possible warranted time. If $Y \subseteq X$, the exact match retrieve is required. Such problems are typical for professional applications on industrial automation, biomedicine, crime prevention, medical diagnosis and prevention, social security, and other multi-spectral computer vision applications.

We shall name elements $[X]_\alpha$, $\alpha \in A$ of power set 2^X as clusters, if they correspond to the partition of set X . Let us consider such partitions that any elements of one cluster do not differ from each other more than on ε , i.e. $\forall x' \neq x''$ we have $[x'] = [x'']$, if $\rho(x', x'') \leq \varepsilon$ and $[x'] \cap [x''] = \emptyset$ otherwise. The given or obtained value ε used at a clustering stage is connected with required accuracy of retrieval δ , if it is specified, as follows. There arise two cases:

$\delta > \varepsilon$ – any representative of the cluster nearest to the query y can be used as the image retrieval result, i.e. minimal number of matching operations is defined by the number of clusters; in other words it is necessary to provide

$$N_1 = \text{card} \{[X]_\alpha\} \rightarrow \min ; \quad (1)$$

$\delta \leq \varepsilon$ – the element of more detailed partition will be the result of the image retrieval. In simplest situations it is necessary to fulfill a single-stage clustering, i.e. to optimize retrieval under worst-case conditions we have to ensure

$$N_2 = \text{card}\{[X]_\alpha\} + \max(\text{card} [X]_\alpha) \rightarrow \min . \quad (2)$$

At the multilevel clustering the repeated clusters search inside of already retrieved clusters is fulfilled and only on the last step required image is searched by complete enumeration. Let us assume that the cluster $[X^{(i-1)}]_p$ is selected on $(i-1)$ level of hierarchy from a condition $\rho(y, [X^{(i-1)}]_q) \rightarrow \min$, $q = \overline{1, \text{card}\{[X^{(i-1)}]\}}$, i.e. $[X^{(i-1)}]_p = [X^{(i)}]_1 \cup [X^{(i)}]_2 \cup \dots \cup [X^{(i)}]_{\alpha_p}$, where for any k and l the equality $[X^{(i)}]_k \cap [X^{(i)}]_l = \emptyset$ holds. Then the minimization of matches amount is reduced to the clustering with the goal function

$$N_3 = \sum_{i=1}^{m-1} \{ \text{card} [X^{(i)}]_{p,(i)} \mid x \in [X^{(i-1)}]_{p,(i-1)} \} + \max (\text{card} [X^{(m-1)}]_{p,(m-1)}) \rightarrow \min , \quad (3)$$

where m is a number of hierarchy levels, $[X^{(0)}]_{1,(0)} = X$. It should be noted, that a new metric to retrieve can be used on every stratum of nested clusters.

3 Hierarchical Partitions with Minimal Matches Amount

To find the multivalued map $\mathfrak{M}_\varepsilon : X \rightarrow \{[X]_\alpha\}$ providing the minimization of (3) let us introduce an iterative procedure $\mathcal{F}_k |_{\mathcal{H}_k(x)} : X \rightarrow 2^X, k = \overline{1, K}$, where

$$\mathcal{H}_k(x) = \mathcal{H}_{k-1}(x) \setminus \mathcal{F}_{k-1}(x), \mathcal{H}_0(x) = X, \mathcal{H}_{K+1}(x) = \emptyset, \mathcal{F}_0(x) = \emptyset, \mathcal{F}_k(x) = \bigcup_{i=1}^{l_k} x_i^*.$$

Here l_k is such that $\mathcal{H}_k^{l_k+1} = \emptyset$, where

$$\mathcal{H}_k^1 = \mathcal{H}_k, \mathcal{H}_k^{i+1} = \{x_i^* \oplus \varepsilon \mathbb{B}^r\} \setminus \{x_i^*\}, x_i^* = \arg \max_{x \in \mathcal{H}_k^i} \text{card}\{x_i \oplus \varepsilon \mathbb{B}^r\}.$$

Notation \oplus designates the Minkowski sum, \mathbb{B}^r is a unit ball in \mathbb{R}^r . The attracting points x_i^* actually give required multivalued map \mathfrak{M}_ε in the form of

$$\mathfrak{M}_\varepsilon = \bigcup_{i=1}^K \mathcal{F}_k. \tag{4}$$

Recall, the value ε is the proximity measure within clusters. Also note that problem (2) is a special case of (3). We proved [21] that union (4) provides the solution of (1), i.e. the map \mathfrak{M}_ε produces partitions (rather, maximal on inclusion posets) $\{[X]_\alpha\}$. Therefore these results can be exploited as initial clusters to get more detailed data partitioning. Finally minimizing (3) is reduced to the search of clusters cardinalities on each hierarchy level.

Let us suppose clusters $\{[X]_\alpha\}$ have cardinalities and multiplicities respectively $(M_1, s_1), \dots, (M_t, s_t), 1 \leq M_1 < M_2 < \dots < M_t \leq \text{card } X$, and the nesting hierarchy number is given (the optimum of hierarchy levels amount will be found further) and it equals m .

Objective function (3) can be rewritten $\forall \tau_1 \in [M_{n-1}, M_n] \cap \mathbb{N} (M_0 = 1)$ as

$$f(\tau_1, \dots, \tau_{m-1}) = \sum_{j=1}^{n-1} s_j + \sum_{j=n}^t \lceil M_j / s_j \rceil s_j + \lceil \tau_1 / \tau_2 \rceil + \dots + \lceil \tau_{m-2} / \tau_{m-1} \rceil + \tau_{m-1}, \tag{5}$$

where values $\tau_1 > \tau_2 > \dots > \tau_m = 1$ correspond to coefficients of sequential clusters partitioning, $\lceil \circ \rceil$ denotes a ceiling function.

Points $\tau_1^*, \tau_2^*, \dots, \tau_{m-1}^*$, at which the global minimum of function $f(\tau_1, \tau_2, \dots, \tau_{m-1})$ is obtained on the set $[1, M_t] \cap \mathbb{N}$, represent required parameters for hierarchical partitions. Local minimum of this function we shall search in the partial segments $[M_{n-1}, M_n]$ ($n = \overline{1, t}$) and global minimum we shall get by search among the obtained values.

For discrete function minimum search we shall first carry out continuous minority function minimizing

$$\varphi(u_1, u_2, \dots, u_{m-1}) = \sum_{j=n}^t M_j s_j / u_1 + \sum_{j=1}^{n-1} s_j + u_1 / u_2 + \dots + u_{m-3} / u_{m-2} + u_{m-1}. \tag{6}$$

Errors of transfer to minority functions are defined by the expression

$$f(u_1, u_2, \dots, u_{m-1}) - \varphi(u_1, u_2, \dots, u_{m-1}) \leq \tilde{\Delta}_n = \sum_{j=1}^{n-1} s_j + m - 2. \tag{7}$$

Consider the function (6) minimizing problem for $u_1 \in [M_{n-1}, M_n]$. Let us emphasize that it has additive form, all items are positive and each item of type u_j/u_{j+1} ($j = \overline{2, m-2}$), evidently, sequentially depends on two variables only. Taking into account these properties, first we can fix u_1, u_2, \dots, u_{m-2} , then we have to find

$$u_{m-2}/u_{m-1} + u_{m-1} \rightarrow \min_{u_{m-1}},$$

whence it follows that $u_{m-1}^* = \sqrt{u_{m-2}}$. Substituting this value into (7) we get

$$\varphi(u_1, u_2, \dots, u_{k-2}) = \sum_{j=n}^t M_j s_j / u_1 + \sum_{j=1}^{n-1} s_j + u_1/u_2 + \dots + u_{m-3}/u_{m-2} + 2\sqrt{u_{m-2}}.$$

Let us find then

$$u_{m-3}/u_{m-2} + \sqrt{u_{m-2}} \rightarrow \min_{u_{m-2}},$$

i.e. $u_{m-2}^* = u_{m-3}^{2/3}$. Continuing this process we come to the relations

$$u_m^* = 1, u_{m-1}^* = (u_{m-2}^*)^{1/2}, \dots, u_2^* = (u_1)^{(m-2)/(m-1)}. \quad (8)$$

Thus we finally arrive at

$$\varphi(u_1) = \varphi(u_1, u_2^*, u_3^*, \dots, u_{k-1}^*) = A_n/u_1 + (k-1)u_1^{1/(k-1)} + B_{n-1},$$

where $A_n = \sum_{j=n}^t M_j s_j$, $B_{n-1} = \sum_{j=1}^{n-1} s_j$.

Let us analyze this function. First notice that $\varphi(u_1)$ is an unimodal function. Indeed, $\varphi'(u_1) = -A_n/u_1^2 + 1/u_1^{(m-2)/(m-1)} = 0$, therefore, $u_1^* = (\sum_{j=n}^t M_j s_j)^{m/(m-1)}$. Further since $\text{sign } \varphi'(u_1) = \text{sign}(-A_n + u_1^{m/(m-1)})$, function $\varphi(u_1)$ decreases on $]0, u_1^*[$ and increases on $]u_1^*, \infty[$. Consequently, minorant (6) reaches its minimum value either at the point u_1^* if $u_1^* \in [M_{n-1}, M_n]$, or at boundary points of this interval. Thereby,

$$\min_{u_1 \in [M_{n-1}, M_n]} \varphi(u_1) = \min \{ \varphi(M_{n-1}), \varphi(M_n), \varphi(\max(\min(u_1^*, M_n), M_{n-1})) \}.$$

Let u_1^{**} be a point at which the minimum is achieved

$$u_1^{**} = \arg \min_{[M_{n-1}, M_n]} \varphi(u_1) \in \{ M_{n-1}, M_n, \max(\min(u_1^*, M_n), M_{n-1}) \},$$

then from (8) we get

$$u_2^* = (u_1^{**})^{(m-2)/(m-1)}; u_3^* = (u_1^{**})^{(m-3)/(m-1)}, \dots, u_{m-1}^* = (u_1^{**})^{1/(m-1)}, u_m^* = 1.$$

Let us find the ranges of definition of u_1, u_2, \dots, u_{m-1} at the return to the discrete goal function. Denote $\psi_i(u_1) = (u_1)^{(m-i)/(m-1)}$ then obviously $\varphi(u_1, u_2, \dots, u_{m-1}) \geq \varphi(u_1, \psi_2(u_1), \psi_3(u_1), \dots, \psi_{m-1}(u_1))$. In the result of minorant minimization we get $\varphi(u_1^*, u_2^*, \dots, u_{m-1}^*) = B^*$. If we choose some value $0 \leq \Delta \leq \tilde{\Delta}_n$ from (7) we have to

solve inequality $\varphi(u_1, u_2, \dots, u_{m-1}) \leq B^* + \Delta$. Denote a solution set as $P(\Delta)$. It is easy to prove that $P(\Delta) \subset P_1(\Delta) \times \psi_2(P_1(\Delta)) \times \dots \times \psi_{m-1}(P_1(\Delta))$.

In that way, finding variables changing intervals $\tau_1, \tau_2, \dots, \tau_{m-1}$ should be started from search of $P_1(\Delta)$. From (7) we find

$$u_1(B^* + \Delta - \sum_{j=1}^{n-1} s_j - (m-1)u_1^{1/(m-1)}) \geq \sum_{j=n}^t M_j s_j. \tag{9}$$

Under $m \geq 4$ inequality (9) can be solved only with numerical methods. Taking into consideration integer-valued character of the search $\tau_1, \tau_2, \dots, \tau_{m-1}$ and relation (7) we shall find the interval $\Delta_1 = [\tau'_1, \tau''_1] \cap \mathbb{N}$, for example, by dichotomy of intervals $[M_{n-1}, \lceil u_1^* \rceil] \cap \mathbb{N}$ and $[\lfloor u_1^* \rfloor, M_n] \cap \mathbb{N}$, ($\lfloor \circ \rfloor$ denotes a floor function) or $[M_{n-1}, M_n] \cap \mathbb{N}$ if $u_1^* \in \{M_{n-1}, M_n\}$. So we finally get

$$\forall j \in \{2, 3, \dots, m-1\} \quad \Delta_j = [\lfloor \psi_m(\tau'_1) \rfloor, \lceil \psi_m(\tau''_1) \rceil] \cap \mathbb{N}.$$

It follows from the above that minimization of retrieval operations number

$$f(\tau_1, \tau_2, \dots, \tau_{m-1}) \rightarrow \min_{[M_{n-1}, M_n]} \tag{10}$$

can be done on the intervals $\tau_1 \in \Delta_1 \cap \mathbb{N}, \tau_2 \in \Delta_2 \cap \mathbb{N}, \dots, \tau_{m-1} \in \Delta_{m-1} \cap \mathbb{N}$.

Let us introduce the hierarchical clusters τ_i -decomposition cortege concept: it is an arbitrary set of integer values $\{\tilde{\tau}_i, \tau_{i+1}, \dots, \tau_{m-1}\}$ satisfying conditions

$$\forall i \in \{1, 2, \dots, m-1\} \Rightarrow \tau_i \in \Delta_i \cap \mathbb{N}, \forall i, j \in \{1, 2, \dots, m-1\}: j \geq i \Rightarrow \tau_j > \tau_{j+1}.$$

Thus, the problem is reduced to the search among all hierarchical clusters τ_i -decomposition vectors such, that the requirement (3) (or (10) what is the same) is met. To solve this problem we shall first draw a backward recurrence of function (5).

The hierarchical clusters τ_i -decomposition cortege we name optimal if under fixed value $\tilde{\tau}_i$ the function

$$F(\tau_{i+1}, \dots, \tau_{m-1}) = \lceil \tilde{\tau}_i / \tau_{i+1} \rceil + \lceil \tau_{i+1} / \tau_{i+2} \rceil + \dots + \lceil \tau_{m-2} / \tau_{m-1} \rceil + \tau_{m-1} \tag{11}$$

has a global optimum on the considered set $(\tilde{\tau}_i, \tau_{i+1}, \dots, \tau_{m-1})$.

On the base of dynamic programming paradigm it is easy to show that the hierarchical clusters τ_j -decomposition cortege belonging to the τ_i -decomposition optimal cortege ($j < i$) is also optimal. Thus we can formulate hierarchical clustering method on the base of backward recurrence, i.e. starting with parameter of τ_{m-1} -decomposition and finishing with τ_1 -decomposition cortege. At each step the concatenation of τ_i -decomposition optimal cortege with an element of higher hierarchical level is carried out. Let us consider this procedure closely.

Under given number of hierarchy levels the input data is a set of integer intervals $\{\Delta_1, \Delta_2, \dots, \Delta_{m-1}\}$. Starting from the lowest level of hierarchy we assume that all the parameters of τ_{m-1} -decomposition are optimal. On the same level we form an initial set $T_{m-1} = \{\tau_{m-1}\}_{\tau_{m-1} \in \Delta_{m-1}}$ of potential optimal decomposition cortege. Then on the

$(m-2)$ level we consider sets $\{\tau_{m-2}, \tau_{m-1}\}$, choosing such $\{\tilde{\tau}_{m-2}, \tau_{m-1}\}$ which provides the minimum of function (11). When we find optimal τ_{m-2} -decomposition corteges, we shall modify the set of potential optimal decomposition corteges $T_{m-2} = \{\tilde{\tau}_{m-2}, \tau_{m-1}\}_{\substack{\tau_{m-2} \in \Delta_{m-2} \\ \tau_{m-1} \in T_{m-1}}}$. Continuing this procedure we obtain

$$\{\tilde{\tau}_i^j, \tau_{i+1}, \dots, \tau_{m-1}\} = \arg \min_{\{\tau_{i+1}, \tau_{i+2}, \dots, \tau_{m-1}\} \in T_{i+1}} \{\tau_i^j \parallel \{\tau_{i+1}, \tau_{i+2}, \dots, \tau_{m-1}\}\},$$

where \parallel denotes a concatenation operation; $T_i = \{\tilde{\tau}_i^j, \tau_{i+1}, \dots, \tau_{m-1}\}$.

The selection of the optimal cortege $\{\tau_1^*, \tau_2^*, \dots, \tau_{m-1}^*\}$ i.e. the set of function (10) arguments is carried out on the last step of the backward recurrence by searching

$$\{\tau_1^*, \tau_2^*, \dots, \tau_{m-1}^*\} = \arg \min_{\{\tau_1, \tau_2, \dots, \tau_{m-1}\} \in T_1} f(\tau_1, \tau_2, \dots, \tau_{m-1}), \quad (12)$$

where the number of operations is within the value $card [\Delta_1 \cap \mathbb{N}]$.

Thus problem (3) solution is obtained in the form (12) on the partial interval $[M_{n-1}, M_n]$ ($n = \overline{1, t}; M_0 = 1$). The global optimum on the whole domain of clustering parameters $[M_1, M_t]$ is chosen among the partial records. So, we have found parameters of clusters partitions at each level of hierarchy. Now there is a need only to divide clusters separately one from other.

It should be emphasized that offered approach enables to use one-parameter sequential optimization on the Cartesian product of multivalued maps $\mathfrak{M}: X \rightarrow \{[X]_\alpha\}$ preimages to search the initial (maximal relative to inclusions) clusters, the degree of objects similarity in these clusters and the stratification coefficient. It can be explained by an independence of the indicated parameters and an absence of principal restrictions to time outlays at preliminary processing of data in CBIR systems.

4 Results and Outlook

Till now we considered clustering with given parameters, namely at known maximum diameter of clusters at solution (1) and amount of strata at solution (3). It is clear that for the reliability growth it is expedient to use sufficiently small values ε , but then a number of matches is increased. Matches number decreasing requires optimization of clusters powers, that is reached by increase of ε . In other words, at $\varepsilon \rightarrow a$, where

$$a = \min \{\lambda \geq 0: (x' \oplus \lambda \mathbb{B}^r) \cap (x'' \oplus \lambda \mathbb{B}^r) \neq \emptyset \forall x', x'' \in X\},$$

the problem is reduced to the exhaustive search. At the same time at $\varepsilon \rightarrow b$, where

$$b = \min \{\lambda \geq 0: X \subseteq x \oplus \lambda \mathbb{B}^r \forall x \in X\},$$

the number of matches also tends to $card X$. The conflict between two criteria (combinatorial complexity of matches and its reliability) is eliminated for each images

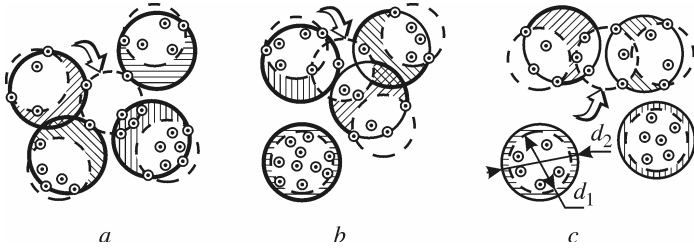


Fig. 1. Clusters merging examples under ϵ changing

configuration at a stage of preliminary processing when there are no key constraints to time outlay. Furthermore to reach desired degree of accuracy and reliability, it is necessary to solve a multiextremal problem. Indeed, fig. 1 illustrates changes ϵ from d_1 to d_2 for problem (2): three cases of clusters amount decreasing are shown (immersed sets are indicated by arrows), but matches number can be increased (a), fixed (b), decreased (c).

Hence the questions concerning rational choice of clustering parameters with respect to CBIR effectiveness are of great significance. A search of optimal ϵ is subject of further inquiry. Here let us find the optimal number of hierarchy levels. By analogy with (6), let us introduce minority function for $[1, M_t]$

$$\varphi_g(u_1, u_2, \dots, u_{m-1}) = \sum_{j=n}^t M_j s_j / u_1 + u_1 / u_2 + \dots + u_{m-3} / u_{m-2} + u_{m-1}.$$

Then from (7) we have $\varphi(u_1, u_2, \dots, u_{m-1}) = m(\sum_{j=n}^t M_j s_j)^{-m} \rightarrow \min_{m \in \mathbb{N}}$, i.e.

$$m \in \{\lfloor L \rfloor, \lceil L \rceil\}, \quad L = \ln(\sum_{j=1}^t M_j s_j).$$

Thus to find optimal number of hierarchy levels it is sufficiently to check only these two values.

The experimental researches of image clustering were carried out in a 29-dimensional feature space. Five groups of features characterizing properties of gray-level functions, their planar peculiarities, objects shapes, invariance to one-parameter geometric transformations and also to Euclidean group of similarities were chosen [21].

As a database a set of images of 108 cars was selected (see fig. 2). Different acquisition conditions were modeled for each image, namely, additive Gaussian and uniform noise, adding the point sources of lighting, discordance of horizontal and vertical scanning, image blurring, image motion modeling, brightness, contrast, lighting intensity variations, linear histogram transformations (see fig. 3). As a result the base set of images was expanded up to 1080 frames. Besides for the image analysis conditioned by variations of a mutual location and/or orientation of object and videosensor, groups of geometric transformations such as rotation, skewing, scaling, perspective distortions, Euclidean similarities (shift, scaling and rotation are acting simultaneously) were used. Thus, the number of the images varied from 1080 up to 6480.

Before clustering all features were normalized to segment $[0,1]$ then they were attributed by weight coefficients. Mentioned above feature families were used either in

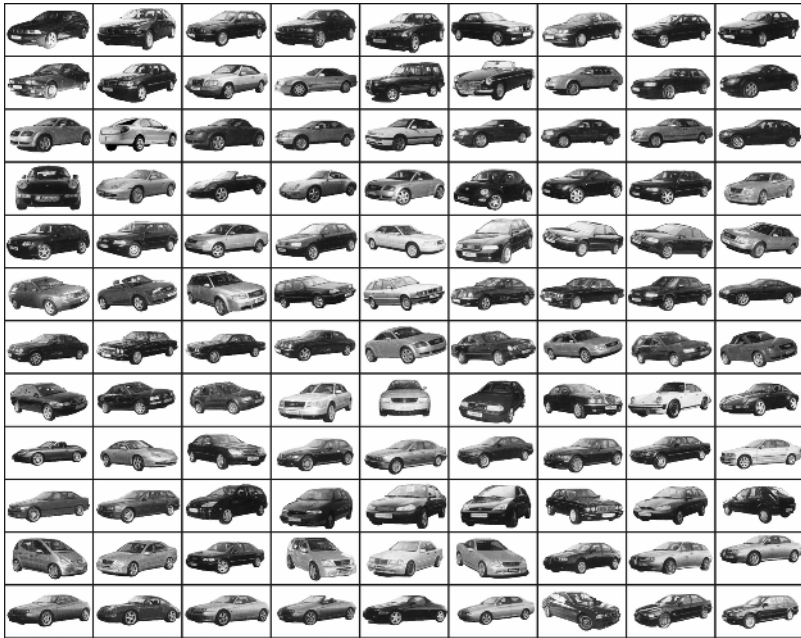


Fig. 2. Basic set of images

total, or separately, or selective by elements. As an example, in fig. 4 the fragment of retrieval via (3) with Hausdorff metric is shown. The analysis of results allows to draw the conclusion that at the correct features selection (namely choice of necessary and sufficient feature set, which has adequate information efficiency) under conditions mentioned number of query matching reduces in 25–110 times as compared to the exhaustive search virtually with idem reliability. In comparison with the most popular hierarchical clustering techniques (nearest and furthest neighbor, median and centroid algorithms, Ward’s minimum variance) matches number is diminished by factor of $N = 6 \dots 17$. It should be emphasized that our approach guarantees minimal number of matches for arbitrary features configuration while traditional clustering methods do it on average. From the practical standpoint the solution of problem (4) often is the most convenient by the virtue of large-scale database processing simplicity. We have ascertained that under minimization (2) the number of matches aspires to $\lambda \sqrt{\text{card } X}$ at essential growth $\text{card } X$ (in ideal case $\lambda = 2$).

To provide fast access to an image in database with queries ‘ad exemplum’ a novel method of hierarchical clustering has been offered and investigated. Proposed method ensures minimal number of matches at CBIR stage for arbitrary images collection in feature or signal space.

In addition we shall indicate that the offered stratified clustering enables to take into account a minimum of clusters diameters, maxima of intercluster or interlevel distances. Moreover we have proposed the clustering method guaranteeing the best

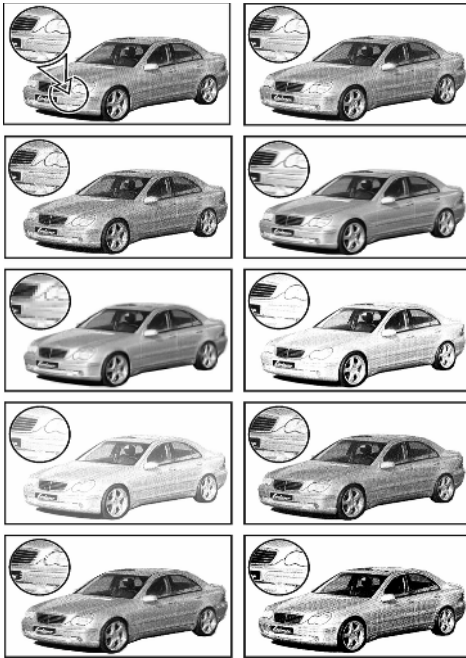


Fig. 3. Variants of image acquisition

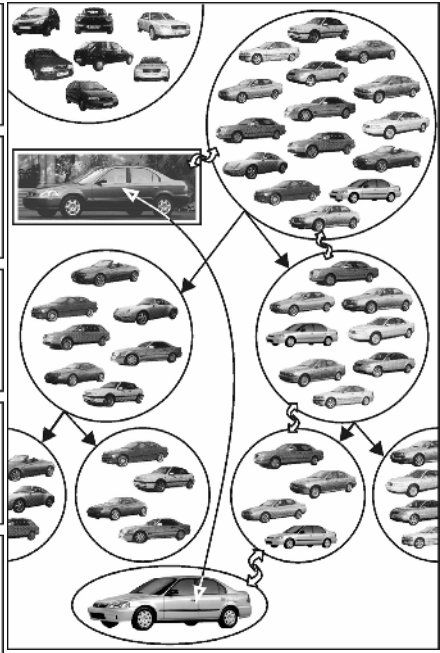


Fig. 4. Fragment of image retrieval

result in a worst-case condition, i.e. often spared hardware-software resource can be used with the purpose of reliability CBIR systems increase. Obeying a key criterion (minimum combinatorial complexity of CBIR), it is also possible to search desired nested clusters by modifications of well-known clustering techniques keeping their advantages.

References

1. Chen, Y., Wang, J.Z.: Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, Vol. 5 (2004) 913-939
2. Li, J., Wang, J.Z.: Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25 (2003) 1075-1088
3. Müller, H., Müller, W., Squire, D.McG., Marchand-Maillet, S., Pun, T.: Performance Evaluation in Content-Based Image Retrieval: Overview and Proposals. *Pattern Recognition Letters*, Vol. 22 (2001) 593-601
4. Veltkamp, R., Burkhardt, H., Kriegel, H.-P. *State-of-the-Art in Content-Based Image and Video Retrieval*. Kluwer Academic Publishers (2001)
5. Greenspan, H., Dvir, G., Rubner, Y.: Context-Dependent Segmentation and Matching in Image Database. *Computer Vision and Image understanding*, Vol. 93, (2004) 86-109

6. Dimai, A.: Assessment of Effectiveness of Content-Based Image Retrieval Systems. *Lecture Notes in Computer Science*, Springer-Verlag, Berlin-Heidelberg, Vol. 1614 (1999) 525-532
7. Rui, Y., Huang, T., Chang, S.: Image Retrieval: Current Techniques, Promising Directions and Open Issues. *Journal of Visual Communication and Image Representation*, Vol. 10 (1999) 39-62
8. Yanai, K., Shindo, M., Noshita, K.: A Fast Image-Gathering System from the World-Wide Web Using a PC Cluster. *Image and Vision Computing*, Vol. 22 (2004) 59-71
9. Bauckhage, C., Braun, E., Sagerer, G.: From Image Features to Symbols and Vice Versa – Using Graphs to Loop Data – and Model-Driven Processing in Visual Assembly Recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 18 (2004) 497-517
10. Manjunath, B.S., Ma, W.Y.: Texture Features for Browsing and Retrieval of Large Image Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18 (1996) 837-842
11. Celebi, E., Alpkocak, A.: Clustering of Texture Features for Content Based Image Retrieval. *Lecture Notes in Computer Science*, Springer-Verlag, Heidelberg, Vol. 1909 (2000) 216-225.
12. Peng, J., Bhanu, B., Qing, S.: Probabilistic Feature Relevance Learning for Content-Based Image Retrieval. *Computer Vision and Image Understanding*, Vol. 75, (1999) 150-164
13. Cox, I.J., Miller, M.L., Minka, T.P., Papatomas, T., Yianilos, P.N.: The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments. *IEEE Transactions on Image Processing*, Vol. 9 (2000) 20-37
14. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Region-Based Image Querying. In: *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries (CVPR' 97)* (1997) 42-49
15. Tian, Q., Sebe, N., Lew, M.S., Loupias, E., Huang, T.S.: Image Retrieval Using Wavelet-Based Salient Points. *Journal of Electronic Imaging*, Vol. 10 (2001) 835-849
16. Cinque, L., De Rosa, F., Lecca, F., Levialdi, S.: Image Retrieval Using Resegmentation Driven by Query Rectangles. *Image and Vision Computing*, Vol. 22 (2004) 15-22
17. Santini, S., Gupta, A., Jain, R.: Emergent Semantics through Interaction in Image Databases. *Knowledge and Data Engineering*, Vol. 13 (2001) 337-351
18. Sheikholeslami, G., Chang, W., Zhang, A.: SemQuery: Semantic Clustering and Querying on Heterogeneous Features for Visual Data. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 14 (2002) 988-1002
19. Käster, T., Wendt, V., Sagerer, G.: Comparing Clustering Methods for Database Categorization in Image Retrieval. *Lecture Notes in Computer Science*, Springer-Verlag, Berlin Heidelberg, Vol. 2781 (2003) 228-235
20. Mashtalir, V.P., Yakovlev, S.V.: Point-Set Methods of Clusterization of Standard Information. *Cybernetics and Systems Analysis*, Kluwer Academic Publishers, Vol. 37 (2001) 295-307
21. Kinoshenko, D., Mashtalir, V., Orlov, A., Yegorova, E.: Method of Creating of Functional Invariants under One-Parameter Geometric Image Transformations. *Lecture Notes in Computer Science*, Springer-Verlag, Berlin-Heidelberg, Vol. 2781 (2003) 68-75

Optimising the Choice of Colours of an Image Database for Dichromats

Vassili Kovalev and Maria Petrou

Centre for Vision, Speech and Signal Processing,
School of Electronics and Physical Sciences,
University of Surrey, Guildford, Surrey GU2 7XH, United Kingdom
{v.kovalev, m.petrou}@surrey.ac.uk

Abstract. Colour appears to gradually play more and more significant role in the modern digital world. However, about eight percent of the population are protanopic and deuteranopic viewers who have difficulties in seeing red and green respectively. In this paper, we identify a correspondence between the 256 standard colours and their dichromatic versions so that the perceived difference between any pair of colours seen by people with normal vision and dichromats is minimised. Colour dissimilarity is measured using the Euclidean metric in the *Lab* colour space. The optimisation is performed using a randomised approach based on a greedy algorithm. A database comprising 12000 high quality images is employed for calculating frequencies of joint colour appearance used for weighting colour dissimilarity matrices.

1 Introduction

Data mining is an active field of research with significant effort being devoted in the recent years into the problem of content-based image retrieval. A large number of such approaches rely on using the colour content of an image as a cue for similarity with a query image [1]. Although colour is not the single most important characteristic which allows one to describe a scene or an object, colour is often being used to attract attention of the viewer to something, to stress something, or even to entice the viewer to a certain product. Colour appears to gradually play more and more significant role in the modern world, and to become a significant part of modern technology, being vital in applications like e-commerce and the digital entertainment industry. And yet, 8% of the population see colours in a totally different way from the rest [2], [3], [4]. These are the people who suffer from some sort of colour blindness, and they usually can distinguish only two hues. They are the protanopic and deuteranopic viewers who have difficulties in seeing red and green respectively. Such people are collectively known as dichromats. A small fraction of people can only see a single hue, and these are the truly colour-blind people [2].

An important issue then arises, concerning the colour world as seen by these viewers, the way it appears to them, and whether the use of colour conveys to them the same information it conveys to normal viewers [5], [3]. Several studies

have been done to answer this question, and indeed we know with pretty high confidence the way the world looks like through the eyes of such viewers (eg [4]). Further, studies have been made in the way colour coded information should be displayed (eg the Paris underground map [5]) so that it is equally useful to all viewers. However, the issue of database search for dichromats using colour as a cue has received much less attention [6]. “An image is a thousand words”, and an image conveys information by the relative colours and contrasts it contains. If a person does not see these contrasts caused by the use of different colours, the person may miss a significant part of the information conveyed by the image. One approach would have been to map all colours in such a way that they will appear as distinct as possible to a dichromat. This, however, might destroy the overall appearance of a picture, as it may create strong contrasts at places where the originator of the picture did not intend them to be. In this paper, we take the view that the perceived difference between the various colours in an image should be preserved by any colour transformation scheme aimed at dealing with the problem of dichromacy. So, we are trying to identify a correspondence between the 256 colours of the standard palette [5], [4], and the 256 colours to which each one of them is transformed by the vision system of the dichromat, so that the perceived difference between any pair of them by a normal viewer and a dichromat viewer is preserved as much as possible. In this work we do not deal with the totally colour blind people who form a very small fraction of the population. Blue-blind people (known as tritanopes), are also extremely rare [2].

At first sight it may seem impossible to map a 3D space to a 2D one and preserve all distances at the same time: The colour space of a normal viewer is 3-dimensional, while the colour space of a dichromat is 2-dimensional, having lost one of the hues. However, one may exploit the differences in perceived saturation to achieve an approximate invariance in perceived difference, and obtain an optimal solution given the constraints of operation. Further, one may consider a weighted approach, where colours that are encountered more often are given more importance than colours that are less frequent. To identify which colours are more frequent we created the normalised 3D colour histogram of each image in a database of natural scenes, and used it as the probability density function for each colour in the standard palette appearing in the image. A cost function was defined, measuring the difference in perceived difference between all possible pairs of colours of the 256 standard colour palette as seen by normals and as seen by dichromats. The perceived difference between any pair of colours was measured using the Euclidean metric in the *Lab* space of normals, and in the *Lab* space of the dichromats [7]. Each term in this cost function was multiplied with a weight which was the minimum value of the probability density function for the two colours as measured from the colour histogram of the corresponding image. This cost function was minimised by using a randomised approach based on a greedy algorithm. This of course does not guarantee the optimal solution, but good sub-optimal solutions could be found. In section 2 we present details of the cost function, the optimisation method, and the image database we used. In section 3 we report the results of our study. Finally, in section 4 we draw our conclusions.

2 Materials and Methods

2.1 Colours Used in This Study

In this study we limit ourselves to the 256 colours considered in [5], [4] together with their dichromatic versions as seen by protanopes and deuteranopes (Figure 1). Construction of dichromatic versions of the colours is based on the LMS specification (the longwave, middlewave and the shortwave sensitive cones) of the primaries of a standard video monitor [8], [9]. The palette of 256 colours we use includes the 216 standard colours that are common for the majority of recent computer applications and computing environments. Conversion from trichromatic to dichromatic colors is done using the *dichromat* package implemented by Thomas Lumley within the R, a language and software environment for statistical computing and graphics [10]-[11].

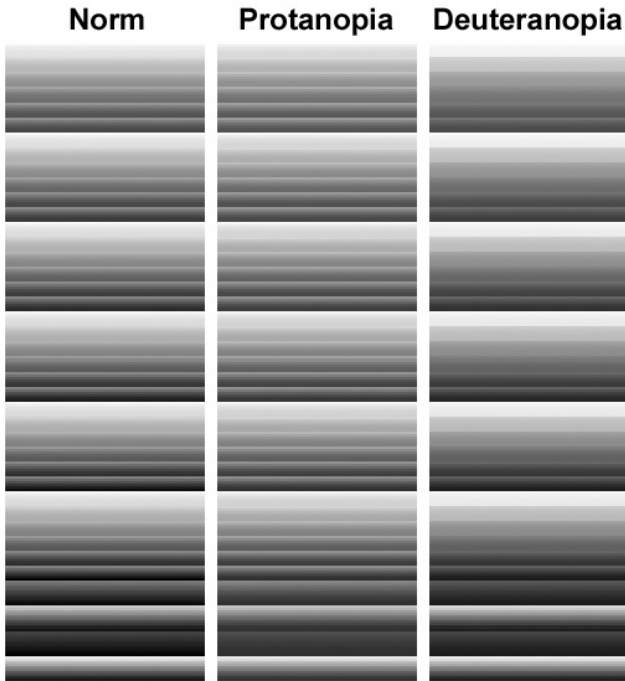


Fig. 1. Palette of 256 colours used in this study as seen by people with normal vision (left column), protanopes (middle column), and deuteranopes (right column)

Note that everywhere in this work colours that differ from those represented in the palette are mapped onto the perceptually closest colours of the palette using the nearest neighbour rule.

2.2 Image Database

The image database we use comprises twelve thousand color RGB images (computer wallpaper photographs) of wide semantic diversity. By convention they are subdivided into 41 categories such as animals, art, aviation, birds, cars, history, food, fantasy, gifts, insects, money, machines, mountains, people, sea, patterns, trains, etc. The original image size of 1024×768 pixels has been reduced by a factor of two to 512×384 for convenience. This database was also used in work [6] concerned with the problem of content-based image retrieval for colour-blind people.

2.3 The Cost Function

The basic idea for improving colour replacement for dichromats is to optimise the mapping of normal colours to their dichromatic versions so that perceived difference between any pair of normal colours is preserved in colour-blind space as much as possible. Let d_{ij}^{NR} and d_{ij}^{CB} be the perceived difference between colours c_i and c_j in the normal and the colour-blind space respectively. Then the cost function formalising the requirement of a best mapping can be written as:

$$U = \sum_{i=1}^{Nc} \sum_{j=1}^{Nc} |d_{ij}^{NR} - d_{ij}^{CB}|,$$

where Nc is the number of colours (dimensionality of colour space), $Nc = 256$. For measuring colour dissimilarity d_{ij} we use the CIE *Lab* colour space [7]. In this space the L component represents the luminance while a and b are the red/blue and yellow/blue chrominances respectively. In the *Lab* space the dissimilarity between the two colours c_i and c_j is computed as the Euclidean distance:

$$d_{ij} = \sqrt{(L_i - L_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2}$$

The cost function U defined above represents the requirement of a minimal deviation of colour dissimilarity observed by dichromats from the colour dissimilarity in subjects with normal vision. However, the colour confusion characteristic for dichromats (see Figure 1) reduces the effective number of distinctive colours, which can be potentially used. Thus, it is worth considering a *weighting* scheme that gives certain emphasis to the colours which appear frequently conjointly in real-world situations:

$$U_w = \sum_{i=1}^{Nc} \sum_{j=1}^{Nc} |d_{ij}^{NR} - d_{ij}^{CB}| \omega_{ij},$$

where ω_{ij} denotes the frequency of joint appearance of colours c_i and c_j .

Note that in both weighted U_w and non-weighted U versions of the cost function we measure the deviation of colour dissimilarities in dichromats from those in subjects with normal vision simply as the sum of absolute values (ie the $L1$ norm) but not as the Euclidean or any other distance metric. This is because the $L1$ norm has been demonstrated to have more consistent behaviour in other similar studies (eg [6], [12], [13]).

2.4 Calculating the Joint Colour Appearance Matrix

Calculating the weighted cost function U_w assumes availability of the frequencies of joint appearance ω_{ij} of all possible pairs of colours c_i and c_j . Clearly, a precise calculation of weighting frequencies ω_{ij} is not possible because it requires analysing a universe of images of real-world scenes. However, since the image database we use is reasonably large and of great content variability, it could provide an acceptable estimate of the frequencies of joint colour appearance. Thus, the estimated frequencies ω_{ij} were calculated in the form of a joint colour appearance matrix, ω , the elements of which represent the frequency of joint appearance of all possible pairs of colours in the images of the database:

$$\omega_{ij} = \sum_{k=1}^{N_{IMG}} \min\{A(c_i^k), A(c_j^k)\},$$

where $A(c_i^k)$ and $A(c_j^k)$ are the values of the normalised 3D colour histogram of the k -th image for colours c_i and c_j respectively, and N_{IMG} is the total number of images.

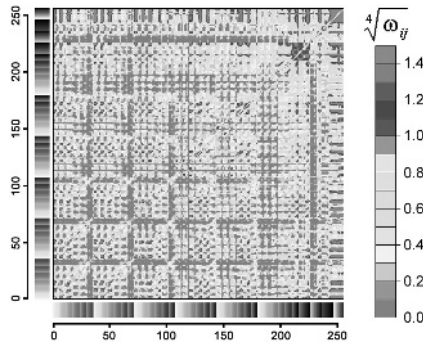


Fig. 2. Matrix of the frequency of joint colour appearance calculated by using the database of twelve thousands images. The matrix element values ω_{ij} are represented by using the non-linear colour scale provided on the right. The high values along the matrix leading diagonal were mapped to the maximum of the remaining values in order to allow the details of the rest of the matrix to be visible.

Figure 2 shows the joint colour appearance matrix calculated using the database. The matrix element values are colour-coded using the colour scale provided on the right. For illustrative purposes, the large values along the matrix leading diagonal were mapped to the maximum of the remaining values in order to allow the details of the rest of the matrix to be visible.

2.5 Optimisation Method

The cost functions U and U_w were minimised by using a randomised approach based on a greedy algorithm. At each iteration step we choose at random an

arbitrary colour c_m^{CB} from the colour-blind palette and replace it by another arbitrary colour c_n^{CB} . The replacement is accepted if it reduces the cost function value. No limitations are applied to the new colour c_n^{CB} except $n \neq m$. This means that colour duplication, colour removal, and transitive restitution of certain colours are possible. The latter operation adds some stochasticity to the algorithm, allowing it to retrace its steps and thus increases its chance to escape from local minima. In general, this algorithm does not guarantee the optimal solution, but good sub-optimal solutions could be found.

The method is implemented using the R language, a free version of S [10] on a P4 3.2GHz PC machine with 2Gb RAM. Depending on the input data, the optimisation procedure takes approximately from 3 to 5 hours to converge.

3 Results

3.1 Optimising Dichromatic Colours Without Weighting

At this stage optimisation was performed without considering the frequency of joint colour appearance. The optimisation procedure was run for protanopes and deuteranopes separately.

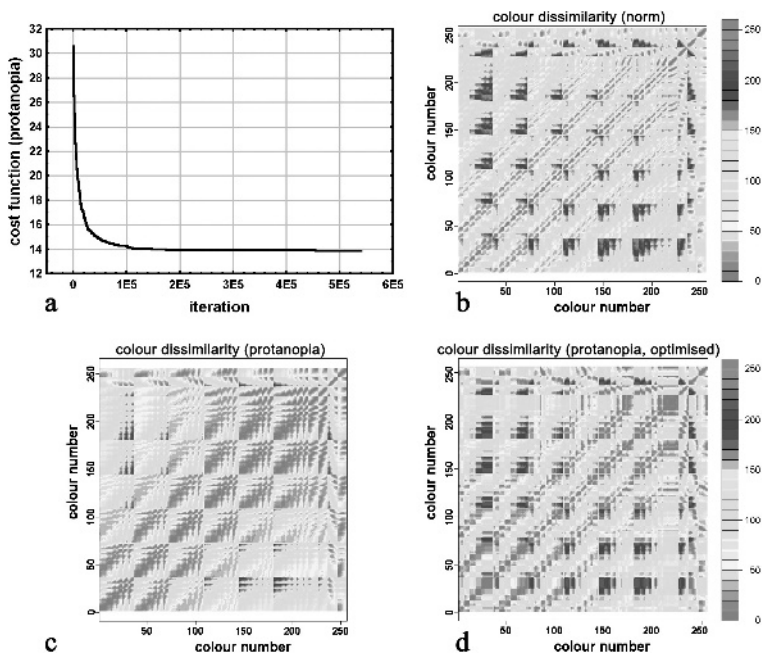


Fig. 3. Optimising protanopic colours without weighting. (a) Changes of the cost function with the iteration steps. (b) Colour-coded representation of the colour dissimilarity matrix for normal vision. (c-d) Colour dissimilarity matrix for protanopia before and after the optimisation. The matrix element values are represented using the linear scale provided on the right

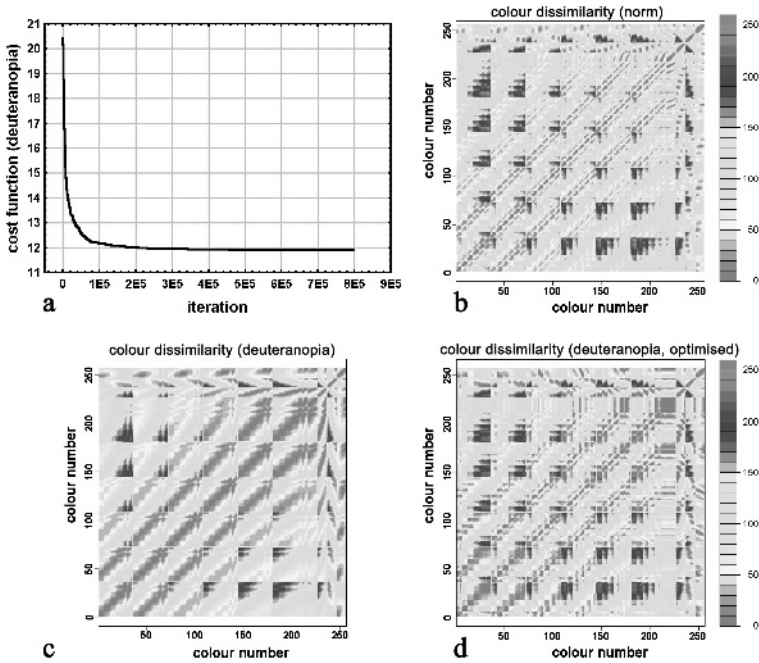


Fig. 4. Optimising deuteranopic colours without weighting. (a) Changes of the cost function with the iteration steps. (b) Colour-coded representation of the colour dissimilarity matrix for normal vision. (c-d) Colour dissimilarity matrix for deuteranopia before and after the optimisation. Matrix element values are represented using the linear scale provided on the right

In case of protanopia (see Figure 3), the initial cost function value $U = 30.49$ dropped down to $U = 14.04$ during the first quarter of the optimisation process ($N = 1.35 \times 10^5$ iterations) and finally converged to the value $U = 13.89$ in $N = 5.4 \times 10^5$ iterations (Figure 3a). These led to substantial changes of the original colour dissimilarity matrix (Figure 3c,d), which became similar to the colour dissimilarity for normal vision (Figure 3b). An iteration in this case is a proposed change of the mapping of colours either it reduces the cost function and so it is accepted, or it does not and so it is rejected.

Optimisation process for the deuteranopic colours (Figure 4) went in a very similar way with slightly slower convergence. As a result, the initial cost function value $U = 20.37$ was reduced almost twice down to $U = 11.92$. In spite of certain differences that can be noticed in the structure of the original colour dissimilarity matrices for protanopia and deuteranopia (see Figure 3c and Figure 4c), the optimised versions were very similar (Figure 3d and Figure 4d).

It should be pointed out that in both occasions the final cost function value remained noticeably different from zero. This is because reduction of the cost function value is proportional to the reduction of the colour gamut in dichromats relatively to normal vision. Thus, converging to zero is not possible in that case.

3.2 Optimising Dichromatic Colours with Weighting

Finally, the optimisation of dichromatic colours was conducted with consideration of the frequencies of joint colour appearance illustrated in Figure 2. The matrix elements ω_{ij} were treated as weights for colour dissimilarities depicted in figures 3c and 4c and the cost function U_w was minimised for both protanopia and deuteranopia.

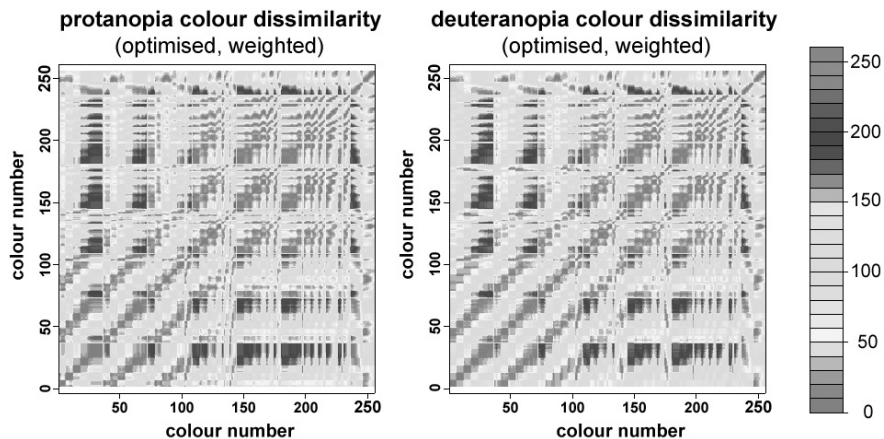


Fig. 5. Colour dissimilarity matrices for protanopia (left) and deuteranopia (right) after colour optimisation with weighting using the frequencies of joint colour appearance ω_{ij}

As a result of the optimisation, the initial cost function value $U_w = 2.38$ was reduced down to $U_w = 1.42$ for protanopic colours and from $U_w = 1.70$ down to $U_w = 1.10$ in case of deuteranopia. During the optimisation process the cost functions behaved similarly to the non-weighted optimisation reported above. Resultant colour dissimilarity matrices for protanopia and deuteranopia are shown in Figure 5. As it can be seen, the optimised dissimilarity matrices depicted in figures 5a and 5b are clearly distinguishable from their non-weighted versions presented in figures 3d and 4d respectively.

The results of colour optimisation using all the ways explored in this study are summarised in Figure 6, where higher colour dissimilarities are evident, after the proposed method is applied.

4 Conclusions

Results reported with this study allow one to draw the following conclusions:

1. The method of optimising the choice of colours of an image database for dichromats suggested in this paper is computationally efficient and able to deal with colour space distortions caused by different kinds of colour deficiency.

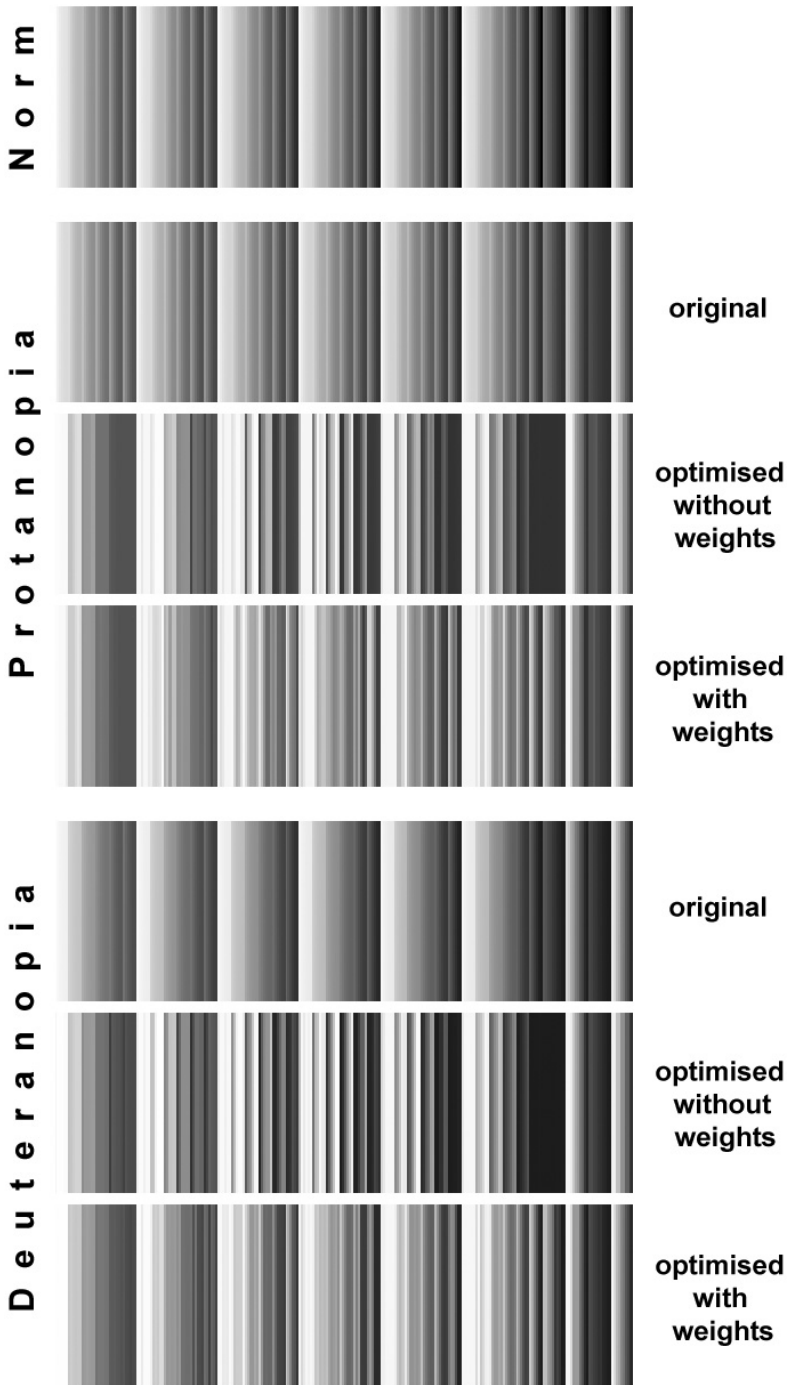


Fig. 6. Results of colour optimisation for all ways explored in this study. Colour palettes which show how much more colours can be distinguishable by colour-blind people after they have been remapped by the proposed methodology

2. Any *a priori* knowledge like statistical information about the frequencies of different colours, and other preferences can be incorporated into the cost function in the form of a weighting matrix.

3. A further study is necessary to investigate the role of an additional optimisation constraint reflecting the (possible) requirement of matching the dichromatic colours with the ones in normal vision in order to minimise psycho-physiological and emotional differences in the perception of real-world scenes.

Acknowledgments

This work was supported by the Basic Technology grant number GR/R87642/01 from the UK Research Council.

References

1. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain., R.: Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Analysis Mach. Intel.* **22** (2000) 1349–1380
2. Viénot, F., Brettel, H., Ott, L., M'Barek, A.B., Mollon, J.: What do color-blind people see? *Nature* **376** (1995) 127–128
3. Rigden, C.: The eye of the beholder - designing for colour-blind users. *British Telecom Engineering* **17** (1999) 2–6
4. Brettel, H., Viénot, F., Mollon, J.: Computerized simulation of color appearance for dichromats. *Journal Optical Society of America* **14** (1997) 2647–2655
5. Viénot, F., Brettel, H., Mollon, J.: Digital video colourmaps for checking the legibility of displays by dichromats. *Color Research Appl.* **24** (1999) 243–252
6. Kovalev, V.A.: Towards image retrieval for eight percent of color-blind men. In: 17th Int. Conf. On Pattern Recognition(ICPR'04). Volume 2., Cambridge, UK, IEEE Computer Society Press (2004) 943–946
7. Hunt, R.W.G.: *Measuring Color*. 2nd edn. Science and Industrial Technology. Ellis Horwood, New York (1991)
8. Meyer, G.W., Greenberg, D.P.: Color-defective vision and computer graphics displays. *IEEE Computer Graphics and Applications* **8** (1988) 28–40
9. Walraven, J., Alferdinck, J.W.: Color displays for the color blind. In: ISandT/SID Fifth Color Imaging Conference: Color Science, Systems and Appl, Scottsdale, Arizona (1997) 17–22
10. Becker, R.A., Chambers, J.M., Wilks, A.R.: *The New S Language*. Chapman and Hall, New York (1988)
11. Maindonald, J., Braun, J.: *Data Analysis and Graphics Using R: An Example-Based Approach*. Cambridge University Press (2003)
12. Kovalev, V., Volmer, S.: Color co-occurrence descriptors for querying-by-example. In: Int. Conf. on Multimedia Modelling, Lausanne, Switzerland, IEEE Computer Society Press (1998) 32–38
13. Rautiainen, M., Doermann, D.: Temporal color correlograms for video retrieval. In: 16th Int. Conf. On Pattern Recognition(ICPR'02). Volume 1., Quebec, Canada, IEEE Computer Society Press (2002) 267–270

An Approach to Mining Picture Objects Based on Textual Cues

Adeoye I. Adegorite, Otman A. Basir, Mohamed S. Kamel,
and Khaled B. Shaban

Pattern Analysis and Machine Intelligence Lab,
Department of Electrical and Computer Engineering,
University of Waterloo, Waterloo,
Ontario N2L 3G1, Canada
{aiadegor, obasir, mkamel, kshaban}@uwaterloo.ca

Abstract. The task of extracting knowledge from text is an important research problem for information processing and document understanding. Approaches to capture the semantics of picture objects in documents constitute subjects of great interest in the domain of document mining recently. In this paper, we present an approach to extracting information about picture objects in a document using cues from the text written about them. The goal of this work is to mine a document and understand the content of picture objects in the document based on meaning inferred from the texts written about such objects. We apply some Natural Language Processing techniques to extract semantic information about picture objects in a document and process texts written about them. The mining algorithms were developed and implemented as a working system and gone through testing and experimentations. Results and future extensions of the work are discussed in this paper.

1 Introduction

The number of electronic documents that contain rich picture objects (PO) has grown enormously in all kinds of information repository, e.g. the World Wide Web (WWW). This growth can be attributed to the increasing use of scanners, digital cameras, and camera-phones in this modern era. Most of these documents contain pictures and texts. Often times, these texts have some cues regarding the contents of the pictures in the document. In the context of this paper, the definition of a PO includes images of different kinds, such as; figures, tables, diagrams, charts, pictures, and graphics. These kinds of picture objects are found in documents databases of medical images, satellite images and digital photographs [1]. Consequently, the option of manually seeking information about POs in a document is highly tedious, particularly when one is dealing with large databases. Thus, there is a need for an efficient mining system that can automatically extract semantically meaningful information about the picture objects from these large document repositories.

There has been some research work focused on either documents mining or image mining separately, in order to extract information from documents [2-5]. The problem addressed in this paper is that of being able to extract information about PO in a document without necessarily carrying out a detailed low-level pixel image mining processes on the PO. The output of this mining system is in form of statements about the PO that are indicative of their contents.

Dixon in [6], has defined document mining as the process of finding interesting or useful patterns in a corpus of textual information. However, image mining deals mainly with the extraction of implicit knowledge, image data relationship, and/or other patterns not directly obvious in the image. Despite the development of many algorithms in these individual research fields, research in image mining is still evolving and at an experimental state [2].

The proposed approach in this paper is to mine the contents of images without performing any low-level pixel/vector based processing. Rather we take advantage of text in the document that reveals some useful information about them.

2 Related Works

There are few related research works that are concerned with the type of problems we are dealing with.

2.1 Mining Multimedia Data

Zaiane *et al.* [7-9], have implemented a prototype for mining high-level multimedia information and knowledge from large multimedia databases. For each image collected, the database would have some descriptive information composed of feature and layout descriptors. The original image is not directly stored in the database; only its feature descriptors are. The descriptive information encompasses fields such as: image file name, image URL, image and video type (i.e. gif, jpeg, bmp, avi, mpeg,), a list of all known web pages referring to the image (i.e. parent URLs), a list of keywords, and a thumbnail used by the user interface for image and video browsing. The image information extractor uses image contextual information, like HTML tags in web pages, to derive keywords. The set of all keywords collected this way, is reduced by eliminating stop-words (e.g., the, a, this) or common verbs (e.g., is, do, have, was), or aggregating words from the same canonical form (e.g., clearing, cleared, clears, clear) as presented in [10].

A drawback of this method is the fact that it is structure-dependent. It relies only on the HTML tags to locate the images in a document. For input files with plain texts without HTML tags, it will be difficult to successfully apply this method.

2.2 Summarization of Diagrams in Documents

Futrelle [11], presented a research work on summarization that attempts to mine information from many diagrams and generates a representative figure that captures all diagrams. The research focused on diagrams, which are line drawings such as

data plots or block diagrams. The overall process is to analysis and to develop structural descriptions. These descriptions are aggregated to produce an all-encompassing structure that summary diagram. One major constraint in this work is the fact that the diagram of interest must be vector-based, as contrasted with normal images, which requires detailed image processing and segmentation in order to analyze them.

Moreover, lexical resources or text accompanying figures were not exploited to guide summarization processes, rather the diagram itself was analyzed by visual parsing. The visual parsing is the only phase that has been reported to be achieved.

3 Text-Based PO Mining

In the following sub-section, we describe the steps involved in mining the contents of PO using the text written about them.

3.1 Systems Procedure and Description

The strategy of information extraction utilized in this project focuses on the PO in a document. Our mining algorithm can be summarized into the following steps:

Step 1: Identify and locate the captions/labels or image tags for each of the PO,taking into consideration the structure of the document. For example, image file name, ALT field in the IMG tag for HTML files can be used to pick the label for any PO.

Step 2: Use the labels or tags obtained in step 1 to derive keywords to search through the document to identify where they appear again in the whole document.

Step 3: Capture the relevant sentences in which the captions/labels already located in step 2 are utilized to further describe or explain the PO.

Step 4: Combine all the sentences captured in step 3 for each PO.

Step 5: Output the statements for each PO.

To better illustrate how these implementation steps were achieved in this project, figure 1 shows a further breakdown of specific processes carried out on each input file. Each of these modules are further explained below:

Input File: The input file to our mining system is the text. The actual PO are not included in the input file, only their captions/labels/image tags and all the text in the document are included.

Pre-processing: The pre-processing tasks done here is tokenization and Part-Of-Speech tagging. The tokenization converts the input file to tokens or units of alphabetic, numeric, punctuation, and white-space charaters. Part-Of-Speech(POS) tagging is done according to the grammatical context of the words in the sentences.

Syntactic Analysis: The purpose of syntactic analysis is to determine the structure of the input text. Here, a tree structure is formed with one node for each phrase. This

structure consists of a hierarchy of *phrases*, the smallest of which are the *basic symbols* and the largest of which is the *sentence*.

Semantic Analysis: This is the method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text. The underlying idea is that the aggregation of all the word contexts in which a given word does and does not appear provides a set of mutual constraints that largely determines the similarity of meaning of words and sets of words to each other.

Information Extraction: Our contribution to this research area is in the strategy utilized at the stage of information extraction. Mining contents of PO is carried out by

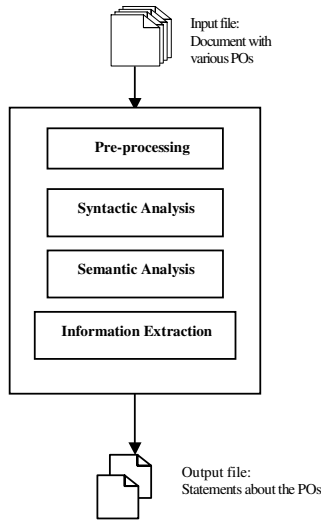


Fig. 1. Text-Based PO Mining

using captions, labels, image file name and ALT tags to search for relevant information about the PO. Sub-section 3.2 gives summarized algorithmic steps of the implementation.

Output File: The output of our system is in form of statements about each PO.

3.2 Mining Algorithmic Steps

Let $X = \{x_1, x_2, \dots, x_n\}$ denote a set of PO in a document $D = \{s_1, s_2, \dots, s_m\}$, where s_1, \dots, s_m are sentences in the document.

Let $Y = \{y_1, y_2, \dots, y_n\}$ denote the caption of the PO and is a subset of D , where $y_i = (l_i, f_i, c_i)$ and $l_i = \text{ALT}\langle \text{label} \rangle$, $f_i = \text{image file name}$ and $c_i = \text{label or title}$.

Let $Z = \{z_{y1}, z_{y2}, \dots, z_{yn}\}$ denote set of relevant sentences that contain captions (y_1, y_2, \dots, y_n) of each PO.

Algorithm 1: Extraction of Information from text about PO

```

1: Input D ← New Document
2: Extract PO tags, captions to X
3: for each  $x_i \in X$  in D do
4:    $y_i \leftarrow$  extract the caption
5:   Add  $y_i$  to Y (List of image-file name and captions for PO)
6:   Z ← Empty List (List of relevant
7:     sentences about each PO)
8:   for each  $y_i$  in Y do
8:     search for sentences  $s_i \in D$  that
       contain  $y_i$ 
9:   If  $y_i$  is a word or phrase in  $s_i \in D$ , then
10:    Add  $s_i$  to  $z_{y_i} \in Z$ 
11:   else
12:    Discard  $s_i$  (irrelevant sentence to PO)
13:   end if
14: end for
15: end for
16: Output the content of Y and Z
  
```

3.3 Systems Architecture

The multi-pass architecture used to implement the proposed algorithm in Visual Text¹ is illustrated in Figure 2.

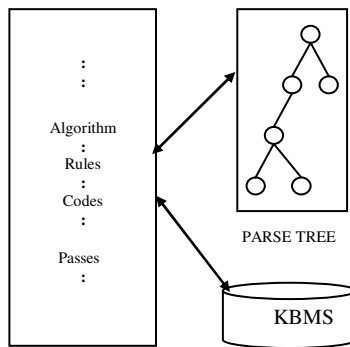


Fig. 2. Text Analyzer Multi-Pass Architecture

Visual Text is an Integrated Development Environment (IDE) that was developed with NLP++ programming language. The passes of the multi-pass architecture are constrained to share a single parse tree. Each pass receives the cumulative parse tree, elaborates it further, and then hands it to a subsequent pass. In addition to managing a unique parse tree, the passes may also update and access a knowledge base, as well as

¹ Visual Text is a trademark of Text Analysis International.

general program data structures. The Visual-Text IDE [14] uses a hierarchical knowledge base management system (KBMS), for mapping knowledge in a more natural fashion than a relational database. The structure of the programming language is a key component of the architecture; that enables the NLP to manage the passes, parse tree and associated knowledge base. By splitting the NLP system into multiple passes, each pass can be constrained to operate on particular contexts. Passes within the architecture can also dynamically create and execute new passes in the processes of tuning the system to get an optimal result.

3.4 Utilized NLP Techniques

In the following sub-sections, we discuss the details of the NLP techniques used in the implementation of this work.

3.4.1 Syntactic Analysis - Parsing

Parsing is the process of linking the part-of-speech tags into a tree structure that indicates the grammatical structure of the sentence. The interior nodes representing phrases, links represent the application of grammatical rules and leaf nodes represent words [13]. Two major types of parsing that are relevant to our system are discussed below:

3.4.1.1 Parsing PCFG

A Probabilistic Context-Free Grammar (PCFG) is a context-free grammar which has a probability associated with each rule normalized so that the probabilities of the rules associated with a particular non-terminal sum to 1. Disambiguation is achieved by selecting the parse tree with the highest probability. The probability of a parse tree, π for a sentence S is given by

$$P(S, \pi) = \prod_{n \in \pi} (P(r(n)))$$

where $p(r(n))$ is probability of the rule r , that has been applied to expand non-terminal n in the parse tree.

3.4.1.2 Lexicalized Parsing

Lexicalized parsers collect two sorts of statistics. Firstly, the probability $P(r|h)$, of which rule r , should be applied given the head h , of the phrase c to be expanded. Secondly, the probability that a sub-phrase q , has head h , given the head of the phrase being expanded m (for “mother”). The total probability for a parse π , of a sentence S is then

$$\rho(S, \pi) = \prod_{c \in \pi} (P(h(c) | m(c))P(r(c) | h(c)))$$

One way of thinking about lexicalized parsers is to imagine them as CFGs (Context-Free Grammar) with a profusion of rules, one for each word in the vocabulary.

3.5 Level of Outputs

We built a multi-pass text analyzer. The analyzer is tuned in terms of number of passes in order to achieve a better result. We identified and established four different

levels of Output. Figure 3 depicts as: Text level, Caption Level, Description level and the Semantic level. The details of these levels are explained as follows:

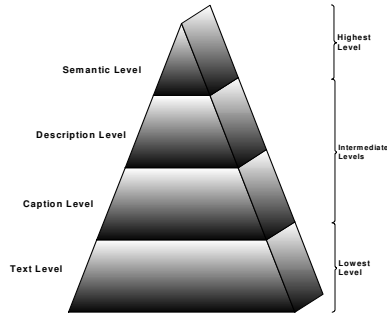


Fig. 3. Levels of Output

- **Text Level:**
This is the state of the input text file at the early part of entry into the Text analyzer. Various passes that have worked on the input file has processed the files into paragraphs and paragraphs into sentences.
- **Caption Level:**
Here, the system has identified and captured the captions and/or labels of the PO.
- **Description level:**
At this stage the mining system has passed through the input file many times, identified the locations of where the figures were referred to in the document and then captured the sentences written about the MMO.
- **Semantic level:**
This is the highest and final level of output. The semantic passes utilize the parse tree and data schemas within the knowledge base. It builds concepts in the knowledge base for sentences, events, and objects in the text that it is processing.

The four information levels can be further generalized to three layers: the text level is the lowest level, while the caption and description level forms the intermediate level and the semantic level is the highest level of output. Figure 4.6 shows these levels.

4 Experimental Results

As indicated in the earlier sub-sections, we built a multi-pass text analyzer. The analyzer is tuned in terms of number of passes in order to achieve an optimal result. After building the text analyzer, many samples of documents were presented to the analyzer. All these documents can be generally categorized into two types. There are some documents with PO that has captions with them, while some documents do not

have captions clearly written with the PO. In the following sub-sections, we present representatives of these two categories and also discuss their results.

4.1 Sample Type 1 – Documents Containing PO with Captions

This is a general case of documents with PO that has captions or labels directly written with the PO. In this case, the text analyzer can easily pick the labels and use this to search through the document to extract sentences that contains these labels. One sample of such documents is shown in figure 4. This document contains many PO, but we have only shown page one, which has four POs. The POs that have captions are written below their corresponding objects. For instance, “Fig 3: Staff of LWF” is shown under the particular PO concerned.

The result obtained from this example is as shown in figure 5 with the respective levels of output.

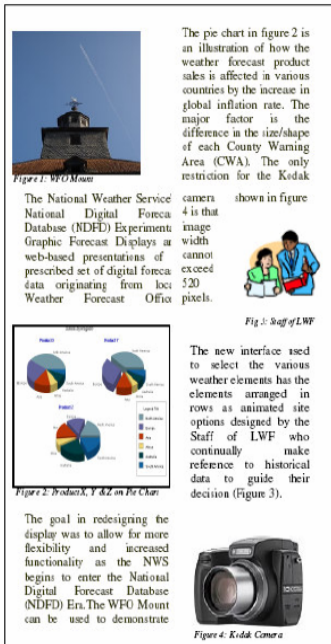


Fig. 4. Sample Document with labeled POs

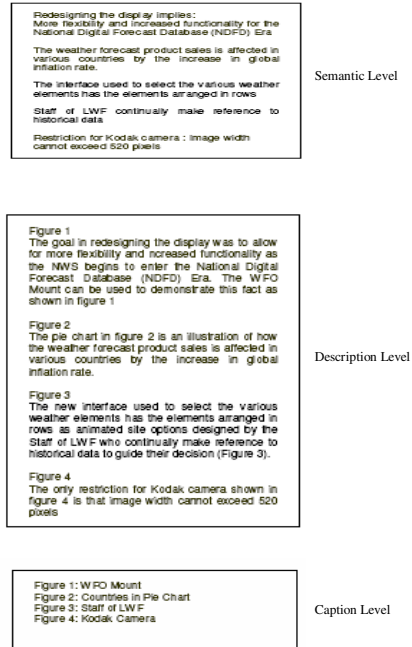


Fig. 5. Levels of Output

4.2 Sample Type 2 – Documents Containing PO Without Captions

This is a general case of documents in which the captions or labels are not found directly written with the PO. In this case, the text analyzer relies on the ALT<label> tab or image file name to pick the labels and use this to search through the document to

extract sentences that contains this labels. A sample document and output is as shown in figure 6 and Figure 7 respectively.

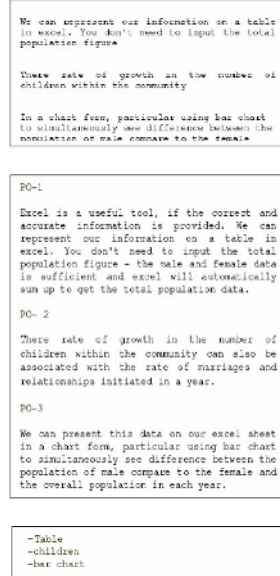
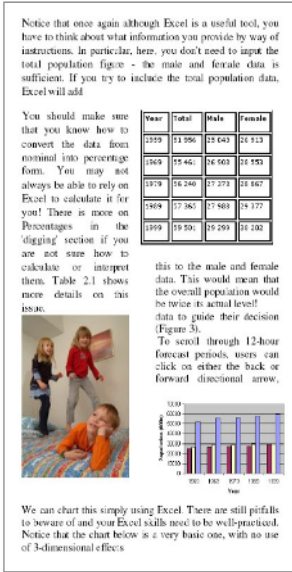


Fig. 6. Sample Document without Labeled POs Fig. 7. Level of Outputs

5 Conclusion and Future Works

This paper presents a research focused on text-based image mining. It is text-based in the sense that, information is extracted from text in the document regardless of the position of the actual wordings in the whole structure. The main advantage of this system is eliminating the detailed image processing that is often suggested by other techniques.

We also established four-level hierarchical structure of output for describing the PO.

The major contribution of this work is the mining algorithm formulated and implemented to produce a text analyzer that can take in an input file in form of text and output statements about the contents of the PO in the document concerned. Comparing our work with some of the related work in the literature [7, 8, 9], the approach we presented is not structure dependent. Another advantage comes in the fact that our approach does not depend on the HTML tags to identify the location of the images. Also, unlike the work done in [11], where diagrams alone are the major focus, our definition of PO is not limited to diagrams, but includes images such as pictures, figures, tables and graphics. Some application areas of this research approach include: Digital library, manuals and indexing. The algorithms developed can also be applied to learning object mining, knowledge representation and knowledge sharing.

The definition of our PO, which encompasses representations in documents such as diagrams, tables, pictures, charts and graphics, implies a constraint boundary, which invariably has limited us to considering only images that falls under these categories. An extension of this research for future work would be to investigate ways of adapting our strategy to capture other types of images in a document, such as video clips, audio clips, flash animated objects, dynamic web-contents and other multimedia objects that may be in any document.

References

1. H. WYNNE, L. L. MONG AND J. ZHANG. "Image Mining: Trends and Developments". In Journal of Intelligent Information System (JISS): Special Issue on Multimedia Data Mining, pp 97-106 Kluwer Academic, 2002.
2. A. POPESCU, L.H. UNGAR, S. LAWRENCE AND D. M. PENNOCK. "Statistical relational learning for document mining". Third IEEE International Conference on Data Mining, ICDM 2003, pp275 – 282, November 19-22, 2003.
3. S. QIN-BAO, L. NAI-QIAN, S. JUN-YI AND C. LI-MING. "Web documents mining". In Proceedings of 2002 International Conference on Machine Learning and Cybernetics, Volume: 2, pp791 – 795 November 4-5, 2002.
4. U. FAYYAD, G. PIATETSKY-SHAPIRO, AND P. SMYTH. "The KDD process for extracting useful knowledge from volumes of data". Communications of the ACM, 39(11): pp27-34, November 1996.
5. H. AHONEN, O. HEINONEN, M. KLEMETTINEN AND A. I. VERKAMO. "Applying Data mining techniques in text analysis" Report C-1997-23, University of Helsinki, Department of Computer Science, March 1997.
6. M. DIXON: "An Overview of Document Mining Technology" A research report Computer Based Learning Unit, University of Leeds, October 1997.
7. S. J. SIMOFF, C. DJERABA AND O. R. ZAIAANE. "MDM/KDD2002: multimedia data mining between promises and problems" ACM SIGKDD Explorations Newsletter Volume 4, Issue 2. pp 118 – 121, December 2002.
8. O. R. ZAIAANE AND S. J. SIMOFF. "MDM/KDD: multimedia data mining for the second time" ACM SIGKDD Explorations Newsletter Volume 3 , Issue 2 COLUMN: Reports from KDD-2001 pp65 – 67, January 2002
9. O. R. ZAIAANE, J. HAN, Z. LI AND J. HOU "Mining Multimedia Data". CASCON'98: Meeting of Minds, pp 83-96, Toronto, Canada, November 1998.
10. O. R. ZAIAANE, A. FALL, R. V. DAHL, AND P. TARAU. "On-line resource discovery using natural language." In Proceedings, RIAO'97 Montreal, Canada, pp65-73, June 25-27, 1997.
11. R. P. FUTRELLE, "Summarization of Diagrams in Documents" In I. Mani & M. Maybury (Eds.), Advances in Automated Text Summarization. Cambridge, MA pp 61-65. March 1999.
12. J. ZHANG, H. WYNNE AND L. L. MONG. "Image Mining: Issues, Frameworks and Techniques" In Second International Workshop on Multimedia Data Mining-MDM/KDD San Francisco, U.S.A., pp 34-42 August 2001,
13. T. K. LANDAUER, P. W. FOLTZ, AND D. LAHAM. "An Introduction to Latent Semantic Analysis". Discourse Processes, (25): pp 259-284, October 1998.
14. A. MEYERS AND D. HILSTER. "Description of the TexUS System as used for MUC-4". Proceedings of MUC-4. DARPA. pp 207-214. March 1992.

Activity and Motion Detection Based on Measuring Texture Change

Longin Jan Latecki¹, Roland Mieziako¹, and Dragoljub Pokrajac²

¹ Temple University, CIS Dept., Philadelphia, PA 19122, USA
{latecki, rmiezian}@temple.edu

² Delaware State University, CIS Dept, Dover, DE 19901, USA
pokie@ist.temple.edu

Abstract. We estimate the speed of texture change by measuring the spread of texture vectors in their feature space. This method allows us to robustly detect even very slow moving objects. By learning a normal amount of texture change over time, we are also able to detect increased activities in videos. We illustrate the performance of the proposed techniques on videos from PETS repository and the Temple University Police department.

1 Introduction

Motion detection algorithms are important research area of computer vision and comprise building blocks of various high-level techniques in video analysis that include tracking and classification of trajectories. It is an obvious and biologically motivated observation that the main clue for detection of moving objects is the changing texture in parts of the view field. All optical flow computation algorithms use derivative computation to estimate the speed of texture change. However, derivative computation may be very unstable in finite domains of images. Therefore, in this paper we introduce a method that does not require any derivative computation. We propose an approach to motion and activity detection based on statistical properties of texture vectors.

Let us focus on a fixed position in a video plane and observe the sequence of texture vectors representing a patch around this position over time. Each texture vector describes the texture of the patch in a single video frame. We assume a stationary camera. If we observe the patch that corresponds to part of the background image, the texture vectors will not be constant due to various factors (e.g., illumination changes, errors of the video capture device), but combined effect is merely a small spread of texture vectors over time. Also a repetitive background motion like tree branches waving in the wind yields a relatively small spread of texture vectors. Since similar texture repeats frequently, the texture vectors in this case are highly correlated.

On the other hand, if a moving object is passing through the observed location, it is very likely that object will have a different texture from the background patch. Therefore, the texture vectors are very likely to have a large spread. Even if different parts of the moving object have the same texture that is the same as the background texture,

the texture vectors will have large spread at the observed location, since different texture parts will appear in the patch. This holds under the assumption that the texture is not completely uniform, since then different texture parts have different texture vectors. To summarize, the proposed approach can identify moving objects even if their texture is identical with the background texture, due to the fact that our classification is based on measuring the amount of texture change and texture structure is extremely unlikely to be perfectly uniform.

Observe that we measure the spread of texture vectors in the texture space. Because of this, we are not able to compute the optical flow directly, i.e., to estimate the directions and speed of moving objects. However, we are able to perform robust detection of moving objects. In comparison to the existing motion detection algorithms [6,7,14], we do not compute any model of the background. We measure the amount of texture change and classify it into two categories: moving and stationary objects. The aforementioned situation in which the background texture and the texture of moving object are similar illustrates a typical situation in which the proposed approach outperforms any background modeling method. In such cases, in the background modeling approaches the texture of a moving object can be easily misclassified as background texture. A detailed explanation follows in Section 3.

Instead of color, gray level, or infrared values at pixel locations, we consider the values of all pixels in spatiotemporal regions represented as 3D blocks. These 3D blocks are represented through compact spatiotemporal texture vectors to reduce the influence of noise and decrease computational demands. In [11] we have shown that the use of such texture vectors in the framework of Stauffer and Grimson [14] can improve the detection of moving objects while potentially cutting back the processing time due to the reduction of the number of input vectors per frame. Thus, we go away from the standard input of pixel values for motion detection that are known to be noisy and the main cause of instability of video analysis algorithms. We stress that the proposed motion detection technique is independent of any particular texture representation used.

To represent texture, we consider the values of all pixels within spatiotemporal regions represented as 3D blocks. A 3D block (e.g., $8 \times 8 \times 3$ block) consists of a few successive frames (e.g., 3) at the same quadratic patch (8×8) of a scene. To compactly represent these values and to reduce the influence of noise, we apply a dimensionality reduction technique by using principal components projection (PCA). As the result, texture is represented by a vector containing only the most significant projected components of texture, while less significant components and noise are filtered out through the process of feature extraction. The most significant projected components represent a small subset of all the projections. The obtained texture vectors provide a compact low-dimensional joint representation of texture and motion patterns in videos and are used as primary inputs to a motion detection technique. As we mentioned above, texture at a given location in video plane is very likely to considerably vary while a moving object is passing through this location. To measure this variance, we estimate covariance matrix of the texture vectors from the same location within a window of a small number of successive frames, and determine the texture spread as the largest eigenvalue of the covariance matrix. This way, we indirectly determine the magnitude of texture variability in the direction of its maximal change. Finally, the decision whether a moving object or a stationary background is identified at a given

spatiotemporal location is made by dynamic distribution learning of the obtained largest eigenvalue.

The proposed technique can use a variety of video sequences as input, ranging from monochromatic gray scale or infra-red (IR) videos to multispectral videos in visible or IR spectral domain. In this paper, we demonstrate the usefulness of the proposed method on several benchmark videos from PETS workshop. The robust performance of the proposed motion detection method, allows us to base our increased activity detection on it. We define motion amount as a sum of motion activities of all blocks in a given frame (Section 4). By applying a simple statistical learning of the motion amount we are able to detect increased activities. We learn the distribution of the total motion amount in all previous frames, under the assumption that mostly normal activities are present. An increased activity is detected as outlier of the learned distribution.

Our approach to increased activity detection does not include any specific domain knowledge about the monitored objects. Such knowledge can be incorporated in our framework, e.g., we can focus on monitoring only human or vehicle activities. By adding a classifier that is able to label moving object categories, we can restrict our attention to particular object categories, e.g., see [18].

A good overview of the existing approaches to motion detection can be found in the collection of papers edited by Remagnino et al. [13] and in the special section on video surveillance in IEEE PAMI edited by Collins et al. [2]. A common feature of the existing approaches for moving objects detection is the fact that they are pixel based. Some of the approaches rely on comparison of color or intensities of pixels in the incoming video frame to a reference image. Jain et al. [7] use simple intensity comparison to reference images so that the values above a given threshold identify the pixels of moving objects. A large class of approaches is based on appropriate statistics of color or gray values over time at each pixel location. (e.g., the segmentation by background subtraction in W4 [6], eigenbackground subtraction [10], etc). Wren et al. [16] were the first who used a statistical model of the background instead of a reference image.

One of the most successful approaches for motion detection was introduced by Stauffer and Grimson [14]. It is based on adaptive Gaussian mixture model of the color values distribution over time at each pixel location. Each Gaussian function in the mixture is defined by its prior probability, mean and a covariance matrix.

The usefulness of dimensionality reduction techniques to compactly represent 3D blocks has already been recognized in video compression. There, 3D discrete cosine and 3D wavelet transforms are employed to reduce the color or gray level values of a large number of pixels in a given block to a few quantized vector components, e.g., [15]. However, these techniques are not particularly suitable for detecting moving objects, since the obtained components do not necessarily provide good means to differentiate the texture of the blocks. Namely, these transformations are context free and intrinsic in that their output depends only on a given input 3D block. In contrast, we propose to use a technique that allows us to obtain an optimal differentiation for a given set of 3D blocks. To reach this goal, we need an extrinsic and context sensitive transformation such that a representation of the given block depends on its context—the set of other 3D blocks in a given video. The Principal Component Analysis (PCA) [8] satisfies these requirements. Namely, for a given set of 3D blocks PCA assigns to

each block a vector of the components that maximize the differences among the blocks. Consequently, PCA components are very suitable to detect changes in 3D blocks.

2 Proposed Methodology

2.1 Video Representation with Spatiotemporal (*sp*) Texture Vectors

We represent videos as three-dimensional (3D) arrays of gray level or monochromatic infrared pixel values $\mathbf{g}_{i,j,t}$ at a time instant t and a pixel location i,j . We divide each image in a video sequence into disjoint $N_{\text{BLOCK}} \times N_{\text{BLOCK}}$ squares (e.g., 8×8 squares) that cover the whole image. Spatiotemporal (3D) blocks are obtained by combining squares in consecutive frames at the same video plane location. In our experiments, we used $8 \times 8 \times 3$ blocks that are disjoint in space but overlap in time, i.e., two blocks at the same spatial location at times t and $t+1$ have two squares in common. The fact that the 3D blocks overlap in time allows us to perform successful motion detection in videos with very low frame rate, e.g., in our experimental results, videos with 2 fps (frames per second) are included.

The blocks are represented by N -dimensional vectors $\mathbf{b}_{I,J,t}$ (e.g., $N=8 \times 8 \times 3$) specified by spatial indexes (I,J) and time instant t . Vectors $\mathbf{b}_{I,J,t}$ contain all values $\mathbf{g}_{i,j,t}$ of pixels in the corresponding 3D block. To reduce dimensionality of $\mathbf{b}_{I,J,t}$ while preserving information to the maximal possible extent, we compute a projection of the normalized block vector to a vector of a significantly lower length $K \ll N$ using a PCA projection matrix $\mathbf{P}_{I,J}^K$ computed for all $\mathbf{b}_{I,J,t}$ at video plane location (I,J) . The resulting *sp* texture vectors $\mathbf{b}_{I,J,t}^* = \mathbf{P}_{I,J}^K \cdot \mathbf{b}_{I,J,t}$ provide a joint representation of texture and motion patterns in videos and are used as input of algorithms for detection of moving objects. We used $K=10$ in our experiments. To compute $\mathbf{P}_{I,J}^K$ we employ the principal values decomposition following [4,5]. A matrix of all normalized block vectors $\mathbf{b}_{I,J,t}$ at video plane location (I,J) is used to compute the $N \times N$ dimensional covariance matrix $\mathbf{S}_{I,J}$. The PCA projection matrix $\mathbf{P}_{I,J}$ for spatial location (I,J) is computed from the $\mathbf{S}_{I,J}$ covariance matrix. The projection matrix $\mathbf{P}_{I,J}$ of size $N \times N$ represents N principal components. By taking only the principal components that corresponds to the K largest eigenvalues, we obtain $\mathbf{P}_{I,J}^K$.

2.2 Detection of Moving Objects by Measuring Texture Spread

The spread of texture vectors over time indicates whether the corresponding object texture is stationary or moving. Recall that each *sp* vector represents texture of the corresponding block. Hence, by observing the characteristics of *sp* vectors change over time, we are able to detect whether a particular block belongs to a moving object or to a background. Consider a single block position in a video plane. We can observe the trajectory of its *sp* vectors, i.e., the loci of *sp* vectors in successive time frames. If during an observed time interval there is no moving object in the block, the *sp* vectors will be close to each other. Hence the variance of *sp* vectors during the time interval will be small. In contrast, if there is a moving object passing through this block, the *sp* texture vectors will change fast, i.e., the *sp* vectors will be spread in the space of their

coordinates. Therefore, the variance of sp vectors within an observation time window will be fairly large. In Fig. 2(a), we show the trajectory of sp vectors corresponding to block location (24,28) in Campus 1 video. To make this visualization possible, we use only first three PCA components for each sp vector. It can be observed that frames when only stationary objects are visible in the observed block location correspond to regions where sp vectors are clustered into fairly spherical shapes (black dots) with small spread. In contrary, when moving objects are passing through this block location, the trajectory of sp vectors (blue-gray dots) is typically elongated and the variance is relatively large.

A simple way to determine the speed of sp vector change would be to compute the norms of their first derivatives. However, computing finite differences of consecutive sp vectors may be unreliable. In order to determine whether the consecutive vectors belong to elongated trajectories, we need to observe whether they are making a consistent progress in one particular direction within a certain time interval. We propose to assess the sp vector spread in the direction of maximal variance. To measure the variance of sp vectors, we compute the covariance matrix of sp vectors corresponding to the same block location for a pre-specified number of consecutive frames. We use the maximal eigenvalue as the measure of trajectory elongation.

More formally, for each location (x,y) , and temporal instant t , we consider vectors

$$b^*_{x,y,t-W}, b^*_{x,y,t-W+1}, \dots, b^*_{x,y,t}, \dots, b^*_{x,y,t+W} . \tag{1}$$

corresponding to a symmetric window of size $2W+1$ around the instant t . For these vectors, we compute the covariance matrix $C_{x,y,t}$. We assign the largest eigenvalue of $C_{x,y,t}$, denoted as $\Lambda_{x,y,t}$ to a given spatiotemporal video position to define a local variance measure, which we will also refer to as *motion measure*

$$mm(x, y, t) = \Lambda_{x,y,t} \tag{2}$$

The larger the motion measure $mm(x,y,t)$, the more likely is the presence of a moving object at position (x,y,t) . An example graph of mm is shown in Fig. 1.

The large values (spikes) correspond to time intervals when moving objects were observed at this particular video location. As this graph suggest, we can label video position (x,y,t) based on the history of $mm(x,y,t)$ values over time (frames 1, ..., $t-1$) as moving by applying an outlier detection method to mm values, i.e., a position is labeled as moving if motion measure value at a given time is classified as outlier.

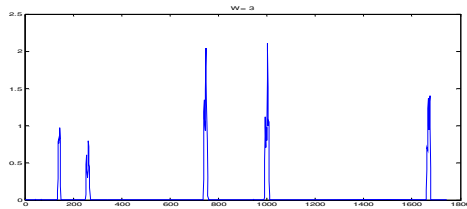


Fig. 1. The graph of local variance mm over time for the block (24,28) of the *Campus 1* video

2.3 Dynamic Distribution Learning and Outlier Detection

In the proposed approach for activity detection we apply outlier detection algorithms two times: for labeling of moving blocks and to detect increased activities. Now we describe outlier detection in more detail. Consider labeling each video position as moving or stationary based on whether the motion measure mm is larger or smaller than a suitably defined threshold. We use a dynamic distribution learning to determine the threshold value at position (x,y,t) based on the history of $mm(x,y,t)$ values over time (at frames 1, ..., $t-1$). Since $mm(x,y,t)$ is a function of one variable t for a fixed position (x,y) (see Fig. 1), the task reduces to dynamic estimation of the mean and standard deviation of mm . Given a function f of one variable, we compute initial values of mean $mean(t_0)$ and variance $\sigma^2(t_0)$ of all values $f(t)$ in some initial interval $t=1..t_0$. An outlier is detected at time $t>t_0$ if the standardized feature value is sufficiently large, i.e., when

$$\frac{f(t) - mean(t-1)}{std(t-1)} > C_1, \text{ where } C_1 \text{ is a constant} \quad (3)$$

Once an outlier is detected at time t_1 , all values $f(t)$ are labeled as outliers for $t_1 < t$ until we switch to a nominal state. We switch to the nominal state at time t , $t_1 < t$, if the standardized feature value drops below a threshold $C_2 < C_1$, i.e.,

$$\frac{f(t) - mean(t-1)}{std(t-1)} < C_2 \quad (4)$$

We update the estimates of mean and standard deviation only when the outliers are not detected (nominal state), i.e., at the beginning of the execution of the algorithm and when (4) holds, $mean$ and std are updated using running average (an algorithm for incremental estimation of parameters of distributions, that is commonly applied in the case of Gaussian distribution):

$$mean(t) = u \cdot mean(t-1) + (1-u) \cdot f(t) \text{ and } std(t) = \sqrt{\sigma^2(t)} \quad (5)$$

$$\sigma^2(t) = u \cdot \sigma^2(t-1) + (1-u) \cdot (f(t) - mean(t-1))^2 \quad (6)$$

For example, we use $C_1=9$, $C_2=3$, and $u=0.99$ in the case of the detection of moving blocks for $f=mm$. The only assumption that we make about the distribution of values of function f is that it has a significant right tail. This assumption clearly applies to the Gaussian distribution, but is significantly more general.

3 Motion Orbits in Texture Space

The most common method to evaluate the performance of motion detection is simply to view the videos with moving objects marked by the applied algorithm as we discuss in Section 2. However, in our framework a more objective method of performance evaluation is also possible. In this section we introduce and use such a method to compare the proposed spread measure of texture vectors to the Gaussian mixture

model introduced in [14]. To make the comparison more realistic, we apply the Gaussian mixture model to texture vectors. Hence, both compared techniques are based on the same spatiotemporal blocks that represent texture and motion patterns. We also show that the Gaussian mixture model on texture vectors significantly outperforms the original representation used in [14] (RGB color values on a pixel level).

We define a motion orbit as path that the texture representation at the fixed video plane location traverses over time. Recall that we use texture vectors composed of the first 3 PCA components of each spatiotemporal block vector. Hence, the motion orbit at video plane location (x,y) is a sequence of points in the 3D Euclidean space $\mathbf{v}_{x,y,1}, \mathbf{v}_{x,y,2}, \dots, \mathbf{v}_{x,y,T}$, where $\mathbf{v}_{l,j,t} = \mathbf{b}_{l,j,t}^*$ and T is the total number of frames.

For instance, in Fig. 2(a), we see the orbit for the block $(24,28)$ of the *Campus 1* PETS video [19]. Frames identified as moving using our local variation method are marked with blue-gray dots while stationary frames are marked with black dots. The distribution of black dots is multimodal globally. We observe two main modes that represent the background blocks. They are identified as two 3D blobs that correspond to two different background textures that appeared in the course of this video at block position $(24,28)$: a part of parking lot and a parked car. Around these blobs we see 1D orbits marked with blue-gray dots corresponding to moving objects. We can view the proposed local variance method as orbit classification algorithm. The reason is that elongated 1D orbits that identify motion have higher spread than the stationary background objects.

We stress that the dot labeling as shown was computed by the proposed method for detection moving objects. Observe that the blue-gray dots perfectly correspond to the 1D motion orbits that identify moving blocks. Thus, our algorithm correctly detected moving objects. In contrast, for the same *Campus 1* video the incremental EM method [14] failed to identify the motion orbit containing frames 633—663. In comparison to any pixel-based approaches (e.g., as originally proposed in [14]), motion detection based on 3D blocks performs better since it reduces noise in background and can extract information about temporal change of texture (since it is based on spatiotem-

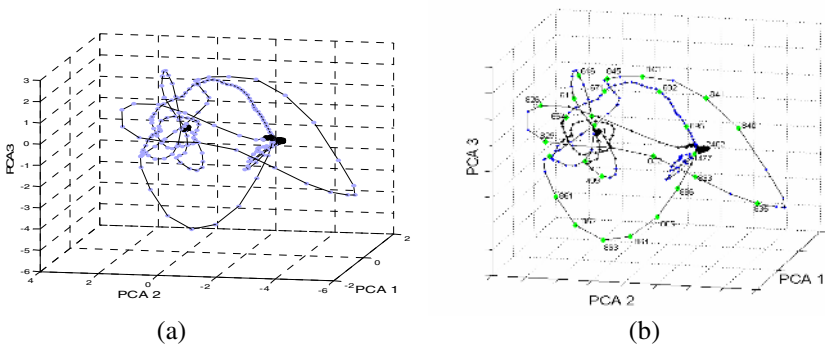


Fig. 2. (a) Orbits of block $(24,28)$ vectors with blue-gray dots corresponding to the frames in *Campus 1* where the block was identified as moving by the proposed method; (b) Orbits of block $(24,28)$ vectors marked with dots: black as background, blue and green as moving—using ‘reset’ and ‘hold’ mechanisms, correspondingly, identified by the EM algorithm

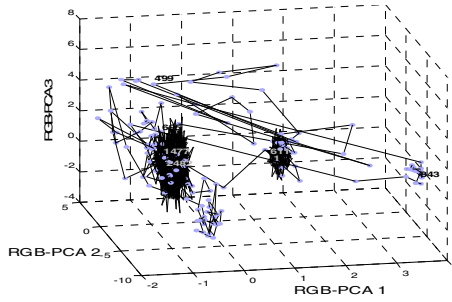


Fig. 3. Standardized PCA components of RGB pixel values for *Campus 1* at pixel location (185,217) that is inside block (24,28); allowS a direct comparison to Fig. 2(a)

poral texture representation of 3D blocks instead of pixels). We demonstrate how noisy RGB color values of a single pixel can be in Fig. 3, where we plot an orbit over time of RGB color values that occur at the pixel (185,217) which is one of the pixels in the block (24,28) of *Campus 1* video. For better visualization, in Fig. 3 we show the linearly transformed space of PCA projections of the original RGB color values (the trajectory in the space of original RGB colors is similar). To allow us a proper comparison to the results in Fig. 2(a) (computed by our local variance technique), we carried over the dot labels from Fig. 2(a).

By comparison of Fig. 3 to Fig. 2(a), one can conclude that in both representations there are two distribution components corresponding to the background. However, using the block-based approach, the background variance is much smaller, since using block vectors that contain texture information results in effective noise reduction in comparison to using “raw” pixels. Hence, any technique to detect moving objects as outliers will perform much better using spatiotemporal blocks than when using the raw pixels. As it can be seen in Fig. 3, the method from [14] have difficulties in properly detecting frames 611, 695, 1477 belonging to the second and fourth moving objects that appear at the observed pixel. The blue-gray dots incorrectly become parts of two background components, which imply that a pixel-based method [14] would classify the corresponding blue-gray dots as belonging to a background distribution. The proposed local variation based technique can also be applied on pixel level. However, due to problems with large uniform texture regions as well as noise inherent to pixel values (shown above), our preferred technique is to apply local variance method on sp block texture vectors.

4 Detection of Increased Activities

Due to the fact that we robustly compute the motion measure mm , we can also reliably estimate the motion amount in each video frame. Motion amount can be defined as the sum of motion measures of all blocks:

$$ma(t) = \sum_{x,y} mm(x, y, t) \quad (7)$$

The proposed method of detecting increased activities is again based on outlier detection (see Section 2.3) but this time of the motion amount over time. Thus, we first learn the distribution of motion amount over time when the recorded video activity was considered usual/nominal. Then time intervals with increased activity are detected as outliers of the learned distribution. The proposed approach works under the assumption that there exists an upper bound on the size of moving objects whose motion we want to detect (measured in the number of moving blocks), and that the genuine moving objects do not appear rapidly in the frame. These assumptions hold for most surveillance videos. Let us consider an example video, called *Temple 1*, that satisfies the assumptions. Indeed, this video is recorded by a roof mounted, stationary camera, so that a certain minimal distance to moving objects is guaranteed. Typical moving objects there, humans and vehicles, cannot get arbitrarily large. Hence, the fraction of the scene occupied by a moving object is limited. Observe that the actual value of the upper bound on the size of moving objects needs not to be known, since our algorithm learns it automatically. Similarly, the number of humans and vehicles cannot rapidly increase, since the regions of entry into the camera view field are limited in size.

In Fig. 4(a), we see the graphs of function ma for *Temple 1* video and correctly detect alarm situations as shown in Fig. 4(b). For example, a significant increase in the

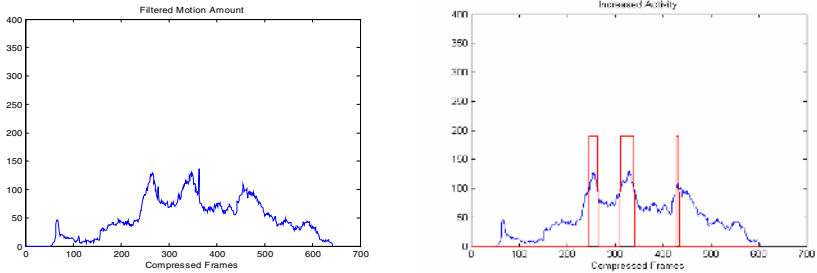


Fig. 4. Activity Detection. (a) Motion amount of *Temple 1* video; (b) Increased activity blocks marked with red boundaries



Fig. 5. *Temple 1* video (a) showing no activity and (b) showing increased activity due to street fight (ACTIVITY label is shown next to the frame number)

number of motion blocks around frame 300 indicates an alarm situation. This is a correct prediction, since a street fight is recorded on the video around frame 300, see Fig. 5 and the *Temple 1* video [12].

5 Performance Evaluation on Test Videos

A set of several test videos showing our motion detection results and our results on detecting increased activity can be viewed on [12]. Our test set of videos includes several videos from the Performance Evaluation of Tracking and Surveillance (PETS) repository. In particular, the results include the above discussed *Campus 1* video from PETS2001, videos obtained from the Police Dept. of Temple Univ., Philadelphia, and infrared videos, for which the same settings of parameters as for visual light videos were used.

6 Conclusions

In this paper we propose a local variation based method for motion detection. Our preliminary results on surveillance and on PETS repository videos show that the proposed method applied to spatiotemporal blocks results in better detection of moving objects in comparison to standard pixel-based techniques and to the incremental EM algorithm technique.

We show that the proposed local variation algorithm can significantly reduce the processing time in comparison to the Gaussian mixture model, due to smaller complexity of the local variation computation, thus making the real time processing of high-resolution videos as well as efficient analysis of large-scale video data viable. Moreover, the local-variation based algorithm remains stable with higher dimensions of input data, which is not necessarily the case for an Gaussian model estimation algorithm. This makes the proposed technique potentially appealing for moving detection in higher dimensional domains, such as multispectral remote sensing imagery.

Our approach to increased activity detection does not include any specific domain knowledge about the monitored objects. Such knowledge can be incorporated in our framework, e.g., we can focus on monitoring only human or vehicle activities if we can restrict our attention to particular object categories.

Acknowledgements

D. Pokrajac has been partially supported by NIH-funded Delaware IDeA Network of Biomedical Research Excellence (INBRE) Grant, DoD HBCU/MI Infrastructure Support Program (45395-MA-ISP Department of Army), National Science Foundation (NSF) Infrastructure Grant (award # 0320991) and NSF grant “Seeds of Success: A Comprehensive Program for the Retention, Quality Training, and Advancement of STEM Student” (award #HRD-0310163).

References

1. Buttler, D., Sridharan, S., and Bove, V. M. Real-time adaptive background segmentation. In Proc. IEEE Int. Conf. on Multimedia and Expo (ICME), Baltimore 2003.
2. R.T. Collins, A.J. Lipton, and T. Kanade, "Introduction to the Special Section on Video Surveillance", IEEE PAMI 22(8) (2000), pp. 745–746.
3. Devore, J. L., Probability and Statistics for Engineering and the Sciences, 5th edn., Int. Thomson Publishing Company, Belmont, 2000.
4. Duda, R., P. Hart, and D. Stork, Pattern Classification, 2nd edn., John Wiley & Sons, 2001.
5. Flury, B. A First Course in Multivariate Statistics, Springer Verlag, 1997.
6. I. Haritaoglu, D. Harwood, and L. Davis, "W4: Real-Time Surveillance of People and Their Activities", IEEE PAMI 22(8) (2000), pp. 809–830.
7. Jain, R., Militzer, D., and Nagel, H. Separating nonstationary from stationary scene components in a sequence of real world TV images. In IJCAI, 612–618, Cambridge, MA, 1977
8. Jolliffe, I. T, Principal Component Analysis, 2nd edn., Springer Verlag, 2002.
9. Javed, O., Shafique, K., and Shah, M. A. Hierarchical approach to robust background subtraction using color and gradient information. In Proc. IEEE Workshop on Motion and Video Computing (MOTION), 22-27, Orlando, 2002.
10. N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions", IEEE PAMI 22(8) (2000), pp. 831–843.
11. D. Pokrajac and L. J. Latecki: Spatiotemporal Blocks-Based Moving Objects Identification and Tracking, IEEE Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), October 2003.
12. Link to Temple ViVi Lab video results. <http://knight.cis.temple.edu/~video/VA/>
13. Remagnino, P., G. A. Jones, N. Paragios, and C. S. Regazzoni, eds., Video-Based Surveillance Systems, Kluwer Academic Publishers, 2002.
14. C. Stauffer, W. E. L. Grimson, "Learning patterns of activity using real-time tracking", IEEE PAMI 22(8) (2000), pp. 747–757.
15. Westwater, R., Furht, B., Real-Time Video Compression: Techniques and Algorithms, Kluwer Academic Publishers, 1997.
16. C. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, "Pfinder: Real-time Tracking of the Human Body", IEEE PAMI 19(7) (1997), pp. 780–785.
17. S. Glisic, Z. Nikolic, D. Pokrajac, P. Leppanen, "Performance Enhancement of DS Spread Spectrum systems: Two Dimensional Interference Suppression," IEEE Trans. Communication, Vol. 47, No. 10, pp.1549-1560, 1999.
18. W. Niu, J. Long, D. Han, and Y.-F. Wang. Human Activity Detection and Recognition for Video Surveillance. In Proc. IEEE Int. Conf. on Multimedia and Expo (ICME), 2004.
19. Performance Evaluation of Tracking and Surveillance (PETS) repository videos Campus 1 and 3: ftp://pets.rdg.ac.uk/PETS2002//DATASET1/TESTING/***/

A New Approach to Human Motion Sequence Recognition with Application to Diving Actions

Shiming Xiang¹, Changshui Zhang¹, Xiaoping Chen², and Naijiang Lu³

¹ State Key Laboratory of Intelligent Technology and Systems,
Department of Automation, Tsinghua University, Beijing 100080, China
{xsm, zcs}@mail.tsinghua.edu.cn

² Department of Physical Education, Tsinghua University, Beijing 100080, China
xpzhq@tsinghua.edu.cn

³ Shanghai Cogent Biometrics Identification Technology Co. Ltd., China
lunj@cbitech.com

Abstract. Human motion sequence-oriented spatio-temporal pattern analysis is a new problem in pattern recognition. This paper proposes an approach to human motion sequence recognition based on 2D spatio-temporal shape analysis, which is used to identify diving actions. The approach consists of the following main steps. For each image sequence involving human in diving, a simple exemplar-based contour tracking approach is first used to obtain a 2D contour sequence, which is further converted to an associated temporal sequence of shape features. The shape features are the eigenspace-transformed shape contexts and the curvature information. Then, the dissimilarity between two contour sequences is evaluated by fusing (1) the dissimilarity between the associated feature sequences, which is calculated by the Dynamic Time Warping (DTW), and (2) the difference between the pairwise global motion characteristics. Finally, sequence recognition is performed according to a minimum-distance criterion. Experimental results show that high correct recognition ratio can be achieved.

1 Introduction

The recent years have seen a surge of interest in video-based human action recognition [1][2][3][4]. However, due to the non-rigidity of human body, human motion classification is a challenging problem. The key difficulty of classification is how to derive the time-varying information from image sequences for action segmentation [4] and motion sequence recognition [3]. Most works [2][4][5] have been done on partitioning an image sequence involving human into key frames, meta-actions, or meta-gestures for video content analysis, human computer interaction, virtual reality, behavior understanding, or sign language recognition. Instead of aiming at analyzing the details within a single image sequence, comparing between different image sequences is desired for intelligent surveillance, content-based video retrieval, video-assisted analysis in athletic training and health-care arenas, and entertainment, etc..

To obtain the motion information from an image sequence, the motion detection and human tracking methods [1][2] can be employed to obtain a sequence of binary silhouettes or a sequence of pose parameters. Since this sequence is associated with the human body, it can reflect the spatio-temporal motion information. We can then derive time-varying feature sequences [3][6] or calculate some important motion properties, such as speed, period, amplitude, number of somersaults in diving, etc.. For example, gait recognition [3] aims to signify the identification of individuals in the image sequences by their gait styles. However, in many applications, identifying who is in an image sequence may be unnecessary. Instead, identifying the motion type to which the motion belongs is desired.

Since gait is a biometric feature, the methods [6] to be used to extract gait feature sequences may not be directly applied to other motions, such as jumping, diving, etc.. Sequence feature analysis for these situations is a new problem.

This paper aims to identify the action group to which the dive belongs. To this end, each image sequence is converted to a 2D contour sequence by our exemplar-based tracking approach. The reasons we analyze 2D contour sequences are: (1) The deformations of the contour can reflect the changes of the pose configuration; (2) Shapes are more robust to the changes of clothing and illumination than color and texture.

The recognition strategy is constructed for the whole 2D contour sequences. We use eigenspace-transformed shape contexts [7] and curvature information as shape features. The features of all contours are listed over time to form a feature sequence. Fig. 1 illustrates the process.

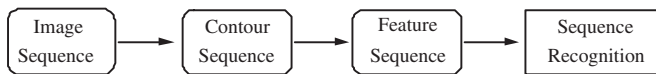


Fig. 1. The process of sequence recognition

Besides the feature sequence, we also use the number of somersaults, which is one of the most distinct global motion characteristics in diving, to describe the 2D contour sequence as a whole. The dissimilarity between two feature sequences is computed by sequence matching through Dynamic Time Warping (DTW) [8] approach. To decide the final dissimilarity between two contour sequences, the dissimilarity of two feature sequences and the difference of global pairwise characteristics are integrated together. Finally, sequence recognition is performed according to a minimum-distance criterion (see Fig. 1).

This paper is structured as follows. Section 2 briefly introduces the related work. Section 3 details the proposed simple and effective approach to deformable contour tracking. The feature analysis approaches to contour sequence are described in Section 4. Section 5 outlines the algorithm of sequence recognition. The experimental results are reported in Section 6, followed the conclusion in Section 7.

2 Related Work

This section briefly reviews the related work on shape representation and motion sequence analysis. The literatures on shape representation are rich [9]. However, we do not need those representations with rotation invariant features, such as Hu moments, Fourier descriptors, and those wavelet based features (see [9] for details), because diving motion is highly related to the rotation of the human body. Whereas, most rotation sensitive representations can only capture the global perception characteristics, for example the spatial moments [9], and hence are incapable of describing the local shape feature well. Belongie et al. proposed a novel method for shape representation and shape matching [7]. The basic idea of their proposal is to construct a shape context for every discretized contour point. Due to the detailed description, measuring the similarity between two points from two shapes can be done explicitly.

From the point view of pattern recognition, two basic tasks are related to image sequence analysis. One task is to partition a sequence into different meta-poses or meta-actions [4]. The other task is to recognize image sequences based on a sequence gallery by taking each of them as a probe sequence. Each probe sequence is described by global motion characteristics or converted to an associated feature sequence. The global characteristics can be derived from time-independent features [10] or time-related features [11]. A feature sequence is a temporal sequence of features, such as the sequences derived from gait styles [3][10][12]. However, different kinds of motions have their own characteristics. Thus, extracting the salient feature is crucial for sequence recognition.

Sequence recognition is performed according to feature comparison. Currently, most of the related works are developed for gait recognition [3][6][10][12]. In contrast, the Hidden Markov Model (HMM) based methods [4][12] and the DTW [8] based methods [3] are more suitable for general sequence comparison. To use HMM, it is necessary to partition the sequences into meta-actions, meta-gestures or key frames as samples to learn the model parameters. The DTW is a common technique since there is no need for one to learn the prior model. However, we need to prepare the sequences to be recognized with roughly equal sequence lengths, according to the work of Rabiner et al. [8].

3 Contour Extraction

We use target tracking approach to extract the contours since the background is non-static. For visual-based human tracking [2], Sequential Monte Carlo (SMC) estimation [13] has proved to be a successful approach. In SMC framework, the probability of the object configuration given the observation is described by a set of weighted particles. Tracking process can then be viewed as a density propagation governed by the dynamic model and observation model [14].

Dynamic model is highly related to contour representation. Due to non-rigid motion and occlusions during diving, representing the 2D deformable diver contours is a tough task. The efficient method with regards to processing defor-

mation is to define complex model with high dimensionality. However, in SMC framework, this leads that the density function which governs the distributions of the target states would be propagated in a high-dimensional state space. It seems that we can use parameterized curves to describe the contour. But due to occlusions of arms, the changes of the 2D pose configuration are drastic.

However, we observe that in diving there exist fundamental poses, which can be used to depict the new ones. To this end, we collect the fundamental contours from different diving action groups to construct a database of exemplars (denoted by E). We use the exemplars to describe the appearances of the target states as well as guide the tracking process. As a result, we can only use three state variables in dynamic model, namely, the centroid coordinate (x, y) and the scale parameter s . Now we can write the dynamic equation as follow:

$$\begin{cases} x_t = x_{t-1} + V_t(x) \\ y_t = y_{t-1} + V_t(y) \end{cases} \tag{1}$$

where (x_t, y_t) is the centroid coordinate of the target state at time t , and $V_t(x)$ and $V_t(y)$ bear normal distribution $N(0, \sigma_x)$ and $N(0, \sigma_y)$, respectively.

Each particle (x_t, y_t) employs an exemplar as its appearance, scaling a little with parameter s . s is randomly set within the range of 0.9-1.1 since the camera was always located in the same place at a distance from the diving platform.

We use the exemplars approximately corresponding to the standing poses to initialize the particles' appearances. After scaled with s , each of them is located in the first frame by using fast Hausdorff distance mapping [15].

Then, we embed a process of contour recognition into the tracking process. For the associated contour of a particle, we retrieval its neighbors from E as its candidates, which are distributed in the current frame according to Equation 1, respectively. After measured through observation model [16], the one with the maximum posterior probability is selected and transferred to the next frame.

To fast retrieval the needed neighbors, the contours in E are organized as a tree structure, based on all the two and three order contour moments.

Toyama et al. perform probabilistic tracking with exemplars in a metric space [17]. There the exemplars are interpreted probabilistically. However, we use exemplars as inputs for searching to find the candidates. Furthermore, we do not need complex training process. Actually, neighbor search approach provides the

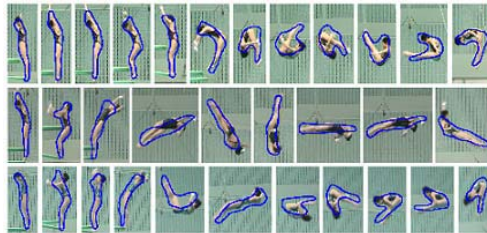


Fig. 2. Some results of contour tracking from three image sequences, respectively

updating dynamics for particles' appearances as well as the mechanism to guide the tracker to find the candidates for each particle. Our method is a simple and effective approach for the purpose of sequence recognition. Fig. 2 shows some tracked frames from three image sequences.

The exemplar database includes 210 different 2D contours. During tracking, the particle number is 4000 and the number of neighbors to be searched is 10. We manually take $\sigma_y = 2\sigma_x$ and $\sigma_x = 8$ since the motion of the centroid of the diver body is roughly controlled by gravity and the motion in the horizontal direction is limited.

4 Sequence Recognition

4.1 Feature Sequence

To convert a contour sequence into a feature sequence, we need the shape features with translation and scale invariance since the contours are translated to the image centers and the body sizes of the divers may be slightly different.

We use shape context descriptor as shape feature. For each reference point, its shape context is a log-polar histogram of the relative coordinates of the remaining points. The shape context summarizes global shape in a rich and local descriptor. Since each point can be associated with a histogram, we can get a shape context matrix, which is a detailed description about the shape perception.

Invariance to translation is intrinsic to the shape context. To achieve scale invariance, all radial distances by the median distance between all the point pairs is normalized [7].

We observe that for most shape contexts a lot of bin values are zeros. This results that the histograms are sparse. Directly using the χ^2 statistics to measure two sparse histograms may not reflect the similarity well [18]. Thus we apply the eigenspace transformation based on Principal Component Analysis (PCA) to the histograms to reduce the redundancy. The details are as follows:

We use all the shape contexts calculated from E as PCA training samples. After performing PCA, we take k eigenvectors corresponding to the k largest eigenvalues, $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_k\}$, to form an eigenspace $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_k]$. For a novel histogram vector \mathbf{X} , we have:

$$\mathbf{Y} = \mathbf{E}^T \mathbf{X} \quad (2)$$

On the other hand, the log-polar space makes the shape descriptor more sensitive to the positions near the reference point. In fact, it is unable to robustly reflect the local geometrical property very well. Actually, the degree of curvature is highly related to a few neighbor points. We use it as an additional feature.

Now a contour is described as a group of features $\{\mathbf{C}, \mathbf{K}\}$, where $\mathbf{C} (\in R^{N \times k})$ is the eigenspace-transformed shape context matrix, and $\mathbf{K} (\in R^N)$ is the curvature vector. Here N is the number of discretized points of the contour and M is the number of the bins of the shape context histogram. As a result, a contour sequence is naturally converted into a feature sequence.

Let point P_S^i belong to contour S , and P_T^j belong to T , the distance between P_S^i and P_T^j can then be computed as follow:

$$d(P_S^i, P_T^j) = \chi^2(\mathbf{C}_S^i, \mathbf{C}_T^j) + s_1 \cdot d_{s2}(\mathbf{C}_S^i, \mathbf{C}_T^j) + s_2 \cdot d_k(\kappa_S^i, \kappa_T^j) + s_3 \cdot d_{k2}(\kappa_S^i, \kappa_T^j) \quad (3)$$

where \mathbf{C}_S^i and κ_S^i denote the eigenspace-transformed shape context and the curvature of the i^{th} point of contour S . \mathbf{C}_T^j and κ_T^j have the same meanings as \mathbf{C}_S^i and κ_S^i , respectively. s_1 , s_2 and s_3 are weighting parameters, which are all manually set as 0.001.

In Formula 3, $\chi^2(P_S^i, P_T^j)$ and $d_{s2}(P_S^i, P_T^j)$ are calculated as the χ^2 statistics and the two order derivative of the eigenspace-transformed shape context cost at the pair point of (P_S^i, P_T^j) [19][20]. $d_k(P_S^i, P_T^j)$ and $d_{k2}(P_S^i, P_T^j)$ are the curvature cost and the two order derivative of the curvature cost, respectively. The reason here we use the two order derivatives is that close points on S should also be close after matched to T .

Finally, the similarity between S and T can be determined, by performing shape matching [7] based on Formula 3.

4.2 Global Motion Characteristics

The number of somersaults (denoted by Π) is a salient global motion characteristic. To calculate Π , we track the position of the feet to form a trajectory and then calculate the rotation number. It is feasible since during diving the feet are always keeping straight and close together and seldom occluded by the arms in the sky. This leads the diver contours have thin appearances. Thus we can extract their skeletons. Now the steps to calculate Π can be summarized as follows:

First, extract the skeleton by morphological thinning operation and trim off the branches with small lengths. Then, detect the branch ends and track the one corresponding to the feet based on the movement continuity. To perform this step, all the vectors defined from the image center to the ends are first normalized. The vector with minimum angle to the tracked vector in the previous frame is selected as current result. Thus, we get a normalized trajectory. Due to translation and normalization, it does not correspond to the real physical one. However, this does not affect the calculation because the diver body turns approximately along its own axis. Finally, Π can be computed as:

$$\Pi = \frac{1}{2\pi} \sum_{i=1}^N \text{sgn}(\mathbf{v}_{i-1} \cdot \mathbf{v}_i) \cdot \arccos(\mathbf{v}_{i-1} \cdot \mathbf{v}_i) \quad (4)$$

where $\text{sgn}(\mathbf{v}_{i-1} \cdot \mathbf{v}_i)$ stands for the relative rotation direction from \mathbf{v}_{i-1} to \mathbf{v}_i , and N the is total frame number. $\text{sgn}(\mathbf{v}_{i-1} \cdot \mathbf{v}_i) = 1$ means the rotation direction is counter-clockwise, while $\text{sgn}(\mathbf{v}_{i-1} \cdot \mathbf{v}_i) = -1$ means the rotation direction is clockwise. Here, \mathbf{v}_0 is the initial position vector.

5 Sequence Recognition Algorithm

Note that the lengths of the image sequences would be very different. To make the sequences to be recognized with almost equal lengths for using DTW matching, we cut off the frames corresponding to the preparing stage because the poses are rest stances and hence weakly informative in the context of action recognition.

To this end, we use again the normalized trajectory. By finding the point which begins to depart from the vertical position, we obtain the corresponding image frame. Thus the sequence can be partitioned into two subsequences. We take the later one for recognition.

The contour sequences may be different every time since the divers may slightly adjust their poses and alter or control the motion speed. Directly performing frame-to-frame matching is not realistic. Therefore, We use the DTW to match the sequences and define the matching cost as dissimilarity [3][8].

Let $S_1 : \{S_1^1, \dots, S_1^n\}$ and $S_2 : \{S_2^1, \dots, S_2^m\}$ be two contour sequences. Let \mathbf{C}_i^j and \mathbf{K}_i^j denote the eigenspace-transformed shape context matrix and the curvature vector of S_i^j , respectively. Suppose the number of somersaults of S_i be Π_i . We summarize the steps of computing the dissimilarity between S_1 and S_2 as follows:

Step 1: Calculate $\mathbf{C}_1^j, \mathbf{K}_1^j$ ($j = 1, \dots, n$) and $\mathbf{C}_2^j, \mathbf{K}_2^j$ ($j = 1, \dots, m$);

Step 2: Transform \mathbf{C}_1^j ($j = 1, \dots, n$) and \mathbf{C}_2^j ($j = 1, \dots, m$), according to Formula 2;

Step 3: Calculate the distance matrix $\mathbf{M} \in (R^{n \times m})$ for S_1 and S_2 :

(1) for S_1^i and S_2^j ($i = 1, \dots, n; j = 1, \dots, m$), compute the pairwise matching cost $e_{i,j}$ based on Formula 3,

(2) let $M_{i,j} = e_{i,j}$;

Step 4: Based on \mathbf{M} , use the DTW matching to calculate the matching cost, and denote it by d_1 ;

Step 5: Calculate Π_1 and Π_2 , and let $d_2 = |\Pi_1 - \Pi_2|$;

Step 6: Compute the dissimilarity between S_1 and S_2 : $d = \sqrt{d_1^2 + (wd_2)^2}$.

Finally, a probe sequence is identified based on the minimum-distance criterion.

6 Experimental Evaluation

The raw video data involving the divers in training was taken at a distance by a CCD camera in different days. To keep the diver figures in the range of the image plane, the camera may slightly rotate along the camera support.

We use the second and fourth group of the international standard diving actions to test our method. The second group action, denoted by ‘2’, is the back group (face to the platform or springboard at the beginning, diving backward). The fourth group (‘4’) is the inward group (diving inward). There have four fundamental pose groups, denoted by ‘A’ (straight), ‘B’ (pike), ‘C’ (tuck) and ‘D’ (free), respectively. The parameters about somersaults are complex. ‘1’ stands



Fig. 3. Some 2D poses and their skeletons extracted by morphological thinning



Fig. 4. Two 2D contour sequences extracted by using the method in Sect. 3



Fig. 5. The translated and normalized trajectories

for ‘0.5’ number of somersaults, ‘2’ for ‘1.0’ number of somersaults, etc. Thus, according to international diving criteria, ‘21A’ means “the second group, 0.5 number of somersaults, straight pose”.

We build a gallery including of 10 groups of diving actions from a single diver: 21A, 23A, 23B, 25B, 23C, 25C, 21D, 23D, 43B, 43C. All the ten image sequences are converted into ten 2D contour sequences by hand. Then a database consisting of 210 different poses is constructed by selecting the same pose once. It is used for both contour tracking (see Sect. 3) and PCA training.

The size of the shape context histogram is 5×12 . The number of the discretized contour points is 80. Thus, there are totally 16800 samples for PCA training. When taking the eigenvectors, we let $k = 20$.

Fig. 3 shows the 2D poses and their branch-trimmed skeletons. Fig. 4 gives two translated sequences. The diving code in Fig. 4(a) is ‘43B’, while the code in Fig. 4(b) is ‘25C’. Fig. 5 demonstrates the translated and normalized trajectories. The trajectories demonstrated in Fig. 5(a) and Fig. 5(b) correspond to the sequences in Fig. 4(a) and Fig. 4(b), respectively. We can see that two sequences have the same length. But they show different pose configuration and different rotation direction (See in Fig. 5). The real values of the numbers of somersaults of the diving actions shown in Fig. 4(a) and Fig. 4(b) are 1.5 and -2.5, respectively. The corresponding values calculated from Formula 4 are 1.45 and -2.38, respectively.

The testing set includes 50 image sequences involving four divers. Each image sequence is used as a probe sequence. The task is to recognize the gallery sequence corresponding to the probe sequence. We use contour tracking to obtain a 2D contour sequence for a probe sequence. The action group of the probe sequence is identified as that of the gallery sequence with which the matching distance is minimum, according to the algorithm in Sect. 5. We achieve 100% correct recognition ratio for 50 testing sequence.

7 Conclusion

This paper aims to recognize diving actions directly based on image sequences. Different from the traditional work on action recognition, we treat the sequence as a whole, rather than partition it into different meta-actions or key frames. We use exemplar-based contour tracking to convert an image sequence into a 2D contour sequence. The eigenspace-transformed shape context histogram matrix and curvature information are used as shape features to form a feature sequence. The dissimilarity of two feature sequences is determined by sequence matching.

The global motion characteristics and motion type determined by the recognition framework of this paper are important video contents for content-based video retrieval and video mining. The meta-data based methods can only summarize the global perception information, which is produced jointly by the humans and the other uninteresting objects.

Although the work is developed on diving actions, the proposed approaches to visual tracking, sequence feature analysis may be applied to other visual computations or video analysis tasks since the related problems are general. In the future, experiments on bigger database and more actions will be carried out to test our method. And we would like to develop more general method for human motion sequences-oriented spatio-temporal pattern analysis.

Acknowledgements

This work is supported by the project (60475001) of the National Natural Science Foundation of China.

References

1. Gavrilu,D.: The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding*, **73** (1999) 82-98
2. Wang, L., Hu, W.M., Tan, T.N.: Recent Developments in Human Motion Analysis. *Pattern Recognition*, **36** (2003) 585-601
3. Wang, L., Tan, T.N., Ning, H.Z., Hu, W.M.: Silhouette Analysis-based Gait Recognition for Human Identification. *Transactions on Pattern Analysis and Machine Intelligence*, **25** (2003) 1505-1518
4. Wu, Y., Huang, T.: Vision-based Gesture Recognition: A Review. In: *Proceedings of the International Gesture Workshop, Gif-sur-Yvette France* (1999) 103-115

5. Cedras, C., Shah, M.: Motion-based Recognition: A Survey. *Image and Vision Computing*, **13** (1995) 129-155
6. Nixon, M.S., Carter, J.N.: Advances in Automatic Gait Recognition. In: *IEEE International Conference on Automatic Face and Gesture Recognition*, Seoul Korea (2004) 139-144
7. Belongie, S., Malik, J., Puzicha, J.: Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24** (2002) 509-522
8. Rabiner, L., Juang, H.: *Fundamentals of Speech Recognition*. Prentice Hall, New Jersey (1993)
9. Loncaric, S.: A Survey of Shape Analysis Techniques. *Pattern Recognition*, **31** (1998) 983-1001
10. Lee, L., Grimson, W.E.L.: Gait Appearance for Recognition. In: *Proceedings of European Conference on Computer Vision*, Copenhagen Denmark (2002) 143-154
11. Collins, R.T., Gross, R., Shi, J.B.: Silhouette-Based Human Identification from Body Shape and Gait. *Proceedings of International Conference of Automatic Face and Gesture Recognition*, Washinton D.C. USA (2002) 351-356
12. Kale, A., Sundaresan, A., Rajagopalan, A. N., et al.: Identification of Humans Using Gait. *IEEE Transactions on Image Processing*, **13** (2004) 1163-1173
13. Doucet, A., De Freitas, N., Ngordon, N.: *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York (2001)
14. Isard, M., Blake, A.: Condensation Conditional- Density Propagation for Visual Tracking. *International Journal of Computer Vision*, **26** (1998) 5-28
15. Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.A.: Comparing Images Using the Hausdorff Distance. *IEEE Transactions on Pattern Analysis and Machine Intellegent*, **15** (1993) 850-863
16. Shen, C.H., van den Hengel, A., Dick, A.: Probabilistic Multiple Cue Integration for Particle Filter Based Tracking. In: *Proceedings of Digital Image Computing: Techniques and Applications*, Sydney Australia (2003) 399-408
17. Toyama, K., Blake, A.: Probabilistic Tracking with Exemplars in a metric Space. *International Journal of Computer Vision*, **48** (2002) 9-19
18. Rubner, Y., Tomasi, C., Guibas, L.J.: The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, **40** (2000) 99-121
19. Thayananthan, A., Stenger, B., Torr, P. H. S., Cipolla, R.: Shape Context and Chamfer Matching in Cluttered Scenes. In: *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Madison Wisconsin (2003) 127-133
20. Srisuk, S., Tamsri, M., Fooprateepsiri, R., Sookavatana, P. Sunat, K.: A New Shape Matching Measure for Nonlinear Distorted Object Recognition. In: *Proceedings of Digital Image Computing: Techniques and Applications*, Sydney Australia (2003) 339-348

Dominant Plane Detection Using Optical Flow and Independent Component Analysis

Naoya Ohnishi¹ and Atsushi Imiya²

¹ School of Science and Technology, Chiba University, Japan
Yayoicho 1-33, Inage-ku, 263-8522, Chiba, Japan
ohnishi@graduate.chiba-u.jp

² Institute of Media and Information Technology, Chiba University, Japan
Yayoi-cho 1-33, Inage-ku, 263-8522, Chiba, Japan
imiya@faculty.chiba-u.jp

Abstract. Dominant plane is an area which occupies the largest domain in an image. A dominant plane detection is an essential task for an autonomous navigation of mobile robots equipped with a vision system, since we assume that robots move on the dominant plane. In this paper, we develop an algorithm for the dominant plane detection using optical flow and Independent Component Analysis. Since the optical flow field is a mixture of flows of the dominant plane and the other area, we separate the dominant plane using Independent Component Analysis. Using an initial data as a supervisor signal, the robot detects the dominant plane. For each image in a sequence, the dominant plane corresponds to an independent component. This relation provides us a statistical definition of the dominant plane. Experimental results using a real image sequence show that our method is robust against a non-unique velocity of the mobile robot motion.

1 Introduction

In this work, we aim to develop an algorithm for a dominant plane detection using the optical flow observed by means of a vision system mounted on a mobile robot. The dominant plane is a planar area which occupies the largest domain in the image observed by a camera. Assuming that robots move on the dominant plane (e.g., floors and ground areas), the dominant plane estimation is an essential task for an autonomous navigation and a path planning of mobile robots.

For the autonomous navigation of mobile robots, vision, sonar and laser sensors are generally used. Sonar and laser sensors [1] provide simple methods of an obstacle detection. These sensors are effective in the obstacle detection for collision avoidance, since these methods can obtain range information to objects. On the other hand, stationary vision sensors have difficulties in obtaining range information. However, vision sensors mounted on a mobile robot can obtain an image sequence from a camera motion. The image sequence provides a motion and structure from correspondences of points on successive images [2]. Additionally, vision sensors are fundamental devices for understanding of an environment,

since robots need to collaborate with human beings. Furthermore, visual information is valid for the path planning of mobile robots in a long sequence, because the vision system can capture environmental information quickly for a large area compared to present sonar- and laser-based systems.

There are many methods for the detection of obstacles or planar areas using vision systems [3]. For example, the edge detection of omni and monocular camera systems [4] and the observation of landmarks [5] are the classical ones. However, since these methods depend on the environment around a robot, they are difficult to apply in general environments. If a robot captures an image sequence of moving objects, the optical flow [6] [7] [8], which is the motion of the scene, is obtained for fundamental features in order to construct environment information around the mobile robot. Additionally, the optical flow is considered as fundamental information for the obstacle detection in the context of biological data processing [9]. Therefore, the use of optical flow is an appropriate method from the viewpoint of the affinity between robots and human beings.

The obstacle detection using optical flow is proposed in [10] [11]. Enkelmann [10] proposed an obstacle-detection method using model vectors from motion parameters. Santos-Victor and Sandini [11] also proposed an obstacle-detection algorithm for a mobile robot using an inverse projection of optical flow to a ground floor, assuming that the motion of the camera system mounted on a robot is pure translation with an uniform velocity. However, even if a camera is mounted on a wheel-driven robot, the vision system does not move with an uniform velocity due to mechanical errors of the robot and a unevenness of the floor.

Independent Component Analysis(ICA) [12] provides a powerful method for texture analysis, since ICA extracts dominant features from textures as independent components [13][14]. We consider optical flow as a texture yielded on surfaces of objects in an environment observed by a moving camera. Therefore, it is possible to extract independent features from flow vectors on pixels dealing with flow vectors as textures. Consequently, we use ICA to separate the dominant plane and the other area.

2 Application of Optical Flow to ICA

ICA [12] is a statistical technique for the separation of original signals from mixture signals. We assume that the mixture signals $x_1(t)$ and $x_2(t)$ are expressed as a linear combination of the original signals $s_1(t)$ and $s_2(t)$, that is,

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t), \quad (1)$$

$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t), \quad (2)$$

where a_{11} , a_{12} , a_{21} , and a_{22} are weight parameters of the linear combination. Using only the recorded signals $x_1(t)$ and $x_2(t)$ as an input, ICA can estimate the original signals $s_1(t)$ and $s_2(t)$ based on the statistical properties of these signals.

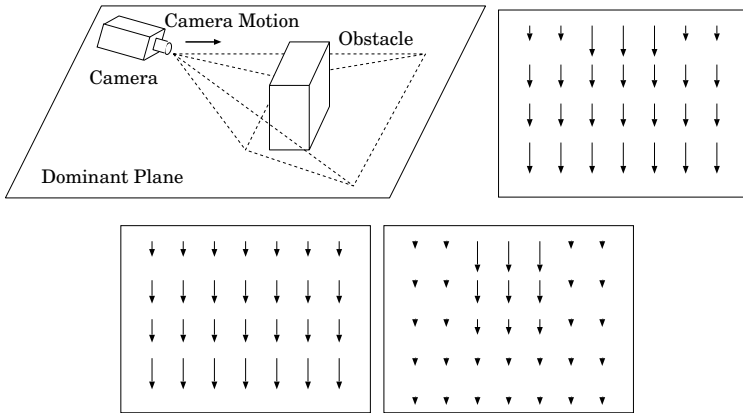


Fig. 1. Mixture property of optical flow. Top-left: Example of camera displacement and the environment with obstacles. Top-right: Optical flow observed through the moving camera. Bottom-left: The motion field of the dominant plane. Bottom-right: The motion field of the other objects. The optical flow(top-right) is expressed as a linear combination of two fields in the bottom

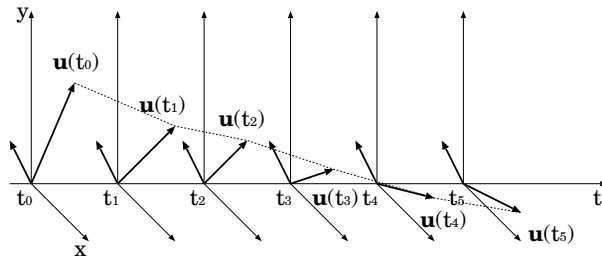


Fig. 2. Dominant vector detection in a sequence of images. $\mathbf{u}(t_i)$ corresponds to the dominant vector which defines the dominant plane at time t_i

We apply ICA to the optical flow observed by a camera mounted on a mobile robot for the detection of the feasible region on which the robot can move. The optical-flow field is suitable as the input to ICA, since the optical flow field observed by a moving camera is expressed as a linear combination of the motion fields of the dominant plane and other objects, as shown in Fig.1. Assuming that the motion field of the dominant plane and other objects are spatially independent components, ICA enables us to detect the dominant plane on which robots can move by separating two signals. For each image in a sequence, we assume that optical flow vectors on pixels in the dominant plane correspond to independent components, as shown in Fig.2.

3 Algorithm for Dominant Plane Detection from Image Sequence

In this section, we develop an algorithm for the detection of the dominant plane from an image sequence observed by a camera mounted on a mobile robot. When the camera mounted on the mobile robot moves on a ground plane, we obtain successive images which include a dominant plane area and obstacles. Assuming that the camera is mounted on a mobile robot, the camera moves parallel to the dominant plane. Since the computed optical flow from the successive images expresses the motion of the dominant plane and obstacles on a basis of the camera displacement, the difference between these optical flow vectors enables us to detect the dominant plane area. The difference of the optical flow is shown in Fig.3.

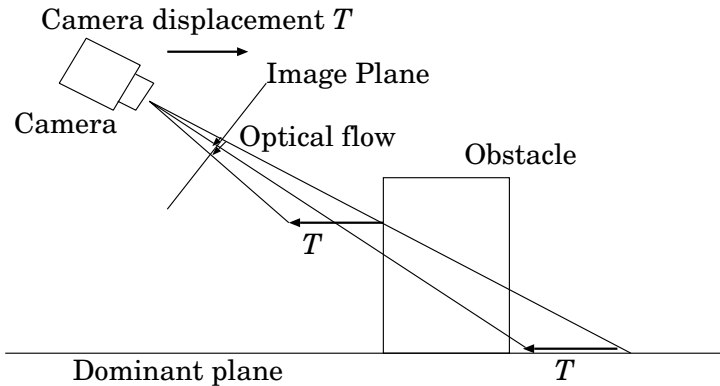


Fig. 3. The difference of the optical flow between the dominant plane and obstacles. If the camera moves in the distance T parallel to the dominant plane, the camera observes different optical flow vectors between the dominant plane and obstacles

3.1 Learning Supervisor Signal

The camera mounted on a robot captures the image sequence $\hat{I}(x, y, t)$ at time t without obstacles as shown in Fig.4 and computes optical flow $\hat{\mathbf{u}}(t) = (\frac{dx}{dt}, \frac{dy}{dt})$ as

$$\hat{\mathbf{u}}(t)^\top \nabla \hat{I}(x, y, t) + \hat{I}_t = 0, \tag{3}$$

where x and y are the pixel coordinates of an image. For the detail of the computation of optical flow $(\frac{dx}{dt}, \frac{dy}{dt})$ using this equation, see [6][7][8].

After we compute optical flow $\hat{\mathbf{u}}(t)$, frame $t = 0, \dots, n - 1$, we create the supervisor signal $\hat{\mathbf{u}}$,

$$\hat{\mathbf{u}} = \frac{1}{n-1} \sum_{t=0}^n \hat{\mathbf{u}}(t). \tag{4}$$

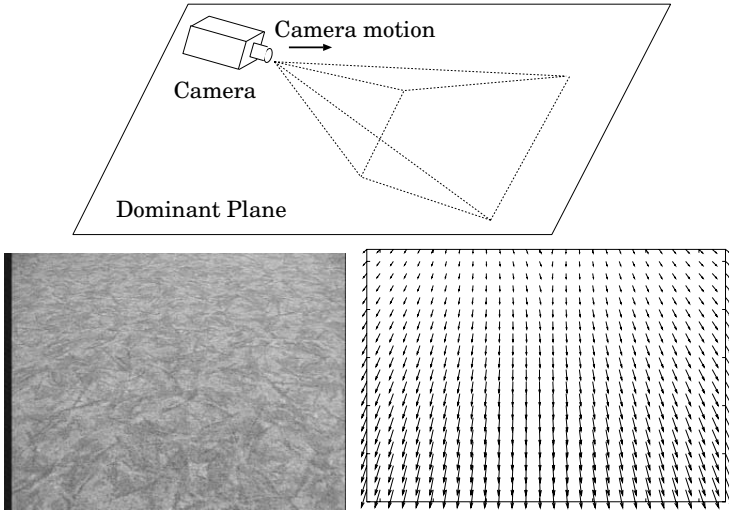


Fig. 4. Captured image sequence without obstacles. Top: Example of camera displacement and the environment without obstacles. Bottom-left: An image of the dominant plane $\hat{I}(x, y, t)$. Bottom-right: Computed optical flow $\hat{\mathbf{u}}(t)$

3.2 Dominant Plane Detection Using ICA

We capture the image sequence $I(x, y, t)$ with obstacles as shown in Fig.5 and compute optical flow $\mathbf{u}(t)$ in the same way.

The optical flow $\mathbf{u}(t)$ and the supervisor signal $\hat{\mathbf{u}}$ are used as input signals for ICA. Setting \mathbf{v}_1 and \mathbf{v}_2 to be output signals of ICA, \mathbf{v}_1 and \mathbf{v}_2 are ambiguity of the order of each component. We solve this problem using a difference between a variance of the length of output signals \mathbf{v}_1 and \mathbf{v}_2 .

Setting \mathbf{l}_1 and \mathbf{l}_2 to be the length of output signals \mathbf{v}_1 and \mathbf{v}_2 ,

$$\mathbf{l}_j = \sqrt{\mathbf{v}_{xj}^2 + \mathbf{v}_{yj}^2}, \quad (j = 1, 2) \tag{5}$$

where \mathbf{v}_{xj} and \mathbf{v}_{yj} are arrays of x and y axis components of output \mathbf{v}_j , respectively, the variance σ_j^2 are

$$\sigma_j^2 = \frac{1}{xy} \sum_{i \in xy} (l_j(i) - \bar{l}_j)^2, \quad \bar{l}_j = \frac{1}{xy} \sum_{i \in xy} l_j(i), \tag{6}$$

where $l_j(i)$ is the i th data of the array \mathbf{l}_j . Since the motions of the dominant plane and obstacles in the image are different, the output, which expresses the obstacle-motion, has larger variance than the output which expresses the dominant plane motion. Therefore, if $\sigma_1^2 > \sigma_2^2$, we detect dominant plane using the output signal \mathbf{l} as $\mathbf{l} = \mathbf{l}_1$, else we use the output signal $\mathbf{l} = \mathbf{l}_2$.

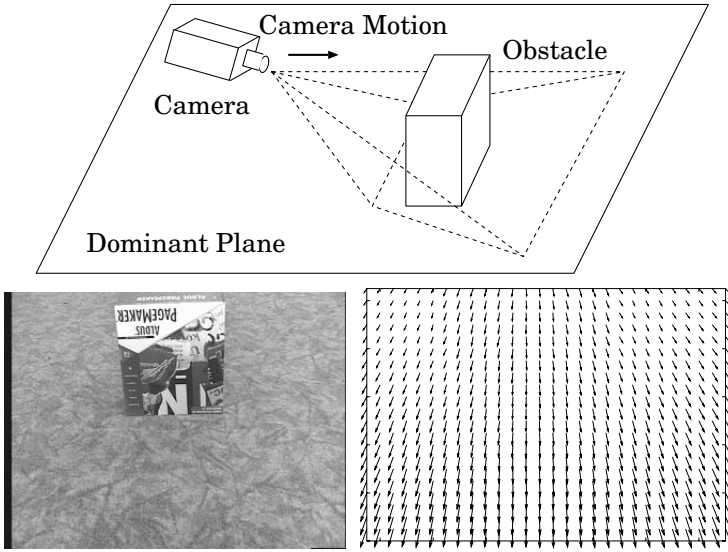


Fig. 5. Optical flow of the image sequence. Top: Example of camera displacement and the environment with obstacles. Bottom-left: An image of the dominant plane and obstacles $I(x, y, t)$. Bottom-right: Computed optical flow $\mathbf{u}(t)$. In a top-middle area, where exists the obstacle, the lengths of optical flow vectors are longer than the flow vectors in the other area

Since the dominant plane occupies the largest domain in the image, we compute the distance between \mathbf{l} and the median of \mathbf{l} . Setting m to be the median value of the elements in the vector \mathbf{l} , the distance $\mathbf{d} = (d(1), d(2), \dots, d(xy))^T$ is

$$d(i) = |l(i) - m|. \tag{7}$$

We detect the area on which $d(i) \approx 0$, as the dominant plane.

3.3 Procedure for Dominant Plane Detection

Our algorithm consists from two phases, learning phase and recognition phase. Learning phase is described as following:

1. Robot moves on the dominant plane in a small distance.
2. Robot captures an image $\hat{I}(u, v, t)$ of the dominant plane.
3. Compute optical flow $\hat{\mathbf{u}}(t)$ between the images $\hat{I}(u, v, t)$ and $\hat{I}(u, v, t - 1)$.
4. If time $t > n$, compute the supervisor signal $\hat{\mathbf{u}}$ using Eq.(4). Else go to step 1.

Next, recognition phase is described as following:

1. Robot moves in the environment with obstacles in a small distance.
2. Robot captures an image $I(u, v, t)$.

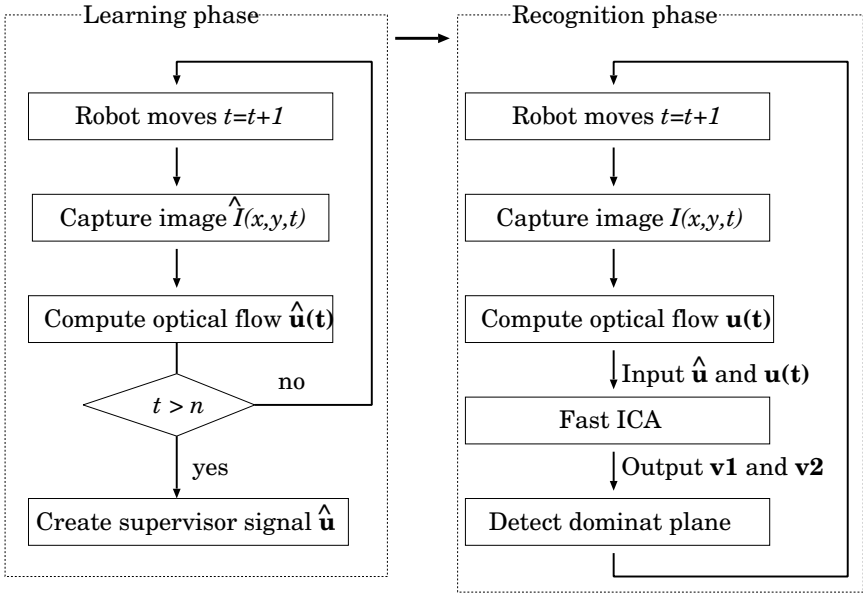


Fig. 6. Procedure for dominant plane detection using optical flow and ICA

3. Compute optical flow $\mathbf{u}(t)$ between the images $I(u, v, t)$ and $I(u, v, t - 1)$.
4. Input optical flow $\mathbf{u}(t)$ and the supervisor signal $\hat{\mathbf{u}}$ to ICA, and output the signals \mathbf{v}_1 and \mathbf{v}_2 .
5. Detect the dominant plane using the algorithm in Section 3.2.

Figure 6 shows the procedure for dominant plane detection using optical flow and ICA.

4 Experiment

We show experiment for the dominant plane detection using the procedure introduced in Section 3.

First, the robot equipped with a single camera moves forward with an uniform velocity on the dominant plane and captures the image sequence without obstacles until $n = 20$. For the computation of optical flow, we use the Lucas-Kanade method with pyramids [15]. Using Eq.(4), we compute the supervisor signal $\hat{\mathbf{u}}$. Figure 7 shows the captured image and the computed supervisor signal $\hat{\mathbf{u}}$.

Next, the mobile robot moves on the dominant plane toward the obstacle, as shown in Fig.8. The captured image sequence and computed optical flow $\mathbf{u}(t)$ is shown in the first and second rows in Fig.9, respectively. Optical flow $\mathbf{u}(t)$ and supervisor signal $\hat{\mathbf{u}}$ are used as input signals for fast ICA. We use the *Fast ICA package for MATLAB* [16] for the computation of ICA. The result of ICA is shown in the third row in Fig.9. Figure 9 shows that the algorithm detects the dominant plane from a sequence of images.

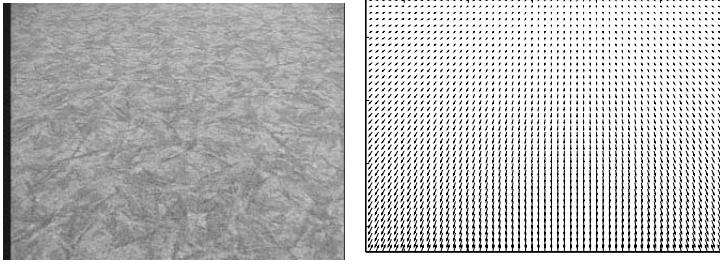


Fig. 7. Doinant plane and optical flow. Left: Image sequence $\hat{I}(x, y, t)$ of the dominant plane. Right: Optical flow \hat{u} used for the supervisor signal



Fig. 8. Experimental environment. An obstacle exists in front of the mobile robot. The mobile robot moves toward this obstacle

For each image in a sequence, the dominant plane corresponds to an independent component. This relation provides us a statistical definition of the dominant plane.

5 Conclusion

We developed an algorithm for the dominant plane detection from a sequence of images observed through a moving uncalibrated camera. The application of ICA to optical flow vector enables the robot to detect a feasible region in which robot can move without requiring camera calibration. These experimental results support the application of our method to the navigation and path planning of a mobile robot equipped with a vision system.

If we project the dominant plane of the image plane onto the ground plane using a camera configuration, the robot detects the movable region in front of the robot in an environment. Since we can obtain the sequence of the dominant plane from optical flow, the robot can move the dominant plane in a space without collision to obstacles. The future work is autonomous robot navigation using our algorithm of the dominant plane detection.

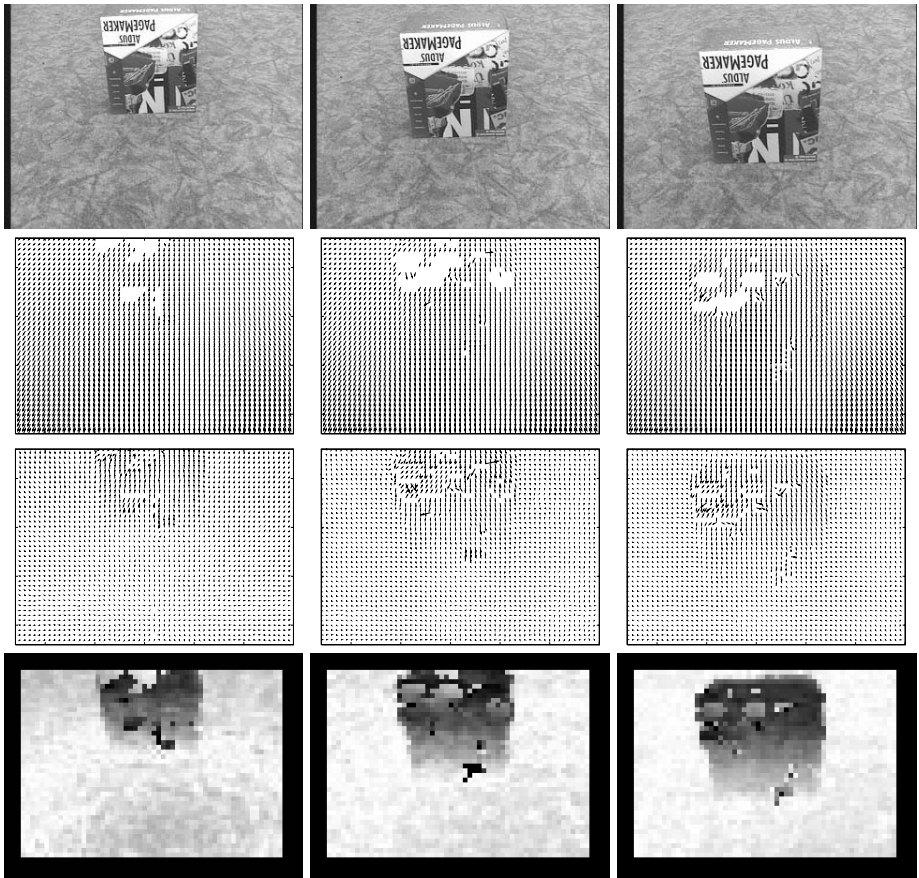


Fig. 9. The first, second, third, and fourth rows show observed image $I(x, y, t)$, computed optical flow $\mathbf{u}(t)$, output signal $\mathbf{v}(t)$, and image of the dominant plane $D(x, y, t)$, respectively. In the image of the dominant plane, the white areas are the dominant planes and the black areas are the obstacle areas. Starting from the left column, $t = 322, 359, \text{ and } 393$

References

1. Hahnel, D., Triebel, R., Burgard, W., and Thrun, S., Map building with mobile robots in dynamic environments, In Proc. of the IEEE ICRA'03, (2003).
2. Huang, T. S. and Netravali, A. N., Motion and structure from feature correspondences: A review, Proc. of the IEEE, **82**, 252-268, (1994).
3. Guilherme, N. D. and Avinash, C. K., Vision for mobile robot navigation: A survey IEEE Trans. on PAMI, **24**, 237-267, (2002).
4. Kang, S.B. and Szeliski, R., 3D environment modeling from multiple cylindrical panoramic images, *Panoramic Vision: Sensors, Theory, Applications*, 329-358, Ryad Benosman and Sing Bing Kang, ed., Springer-Verlag, (2001).

5. Fraundorfer, F., A map for mobile robots consisting of a 3D model with augmented salient image features, 26th Workshop of the Austrian Association for Pattern Recognition, 249-256, (2002).
6. Barron, J.L., Fleet, D.J., and Beauchemin, S.S., Performance of optical flow techniques, *International Journal of Computer Vision*, **12**, 43-77, (1994).
7. Horn, B. K. P. and Schunck, B.G., Determining optical flow, *Artificial Intelligence*, **17**, 185-203, (1981).
8. Lucas, B. and Kanade, T., An iterative image registration technique with an application to stereo vision, *Proc. of 7th IJCAI*, 674-679, (1981).
9. Mallot, H. A., Bulthoff, H. H., Little, J. J., and Bohrer, S., Inverse perspective mapping simplifies optical flow computation and obstacle detection, *Biological Cybernetics*, **64**, 177-185, (1991).
10. Enkelmann, W., Obstacle detection by evaluation of optical flow fields from image sequences, *Image and Vision Computing*, **9**, 160-168, (1991).
11. Santos-Victor, J. and Sandini, G., Uncalibrated obstacle detection using normal flow, *Machine Vision and Applications*, **9**, 130-137, (1996).
12. Hyvarinen, A. and Oja, E., Independent component analysis: algorithms and application, *Neural Networks*, **13**, 411-430, (2000).
13. van Hateren, J., and van der Schaaf, A., Independent component filters of natural images compared with simple cells in primary visual cortex, *Proc. of the Royal Society of London, Series B*, 265, 359-366, (1998).
14. Hyvarinen, A. and Hoyer, P., Topographic independent component analysis, *Neural Computation*, **13**, 1525-1558, (2001).
15. Bouguet, J.-Y., Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm, Intel Corporation, Microprocessor Research Labs, OpenCV Documents, (1999).
16. Hurri, J., Gavert, H., Sarela, J., and Hyvarinen, A., The FastICA package for MATLAB, website: <http://www.cis.hut.fi/projects/ica/fastica/>

Neural Expert Model Applied to Phonemes Recognition

Halima Bahi and Mokhtar Sellami

LRI laboratory, Computer Science Department, University of Annaba,
BP12, 23000 Annaba, Algeria
{bahi, sellami}@lri-annaba.net
<http://lri-annaba.net>

Abstract. Connectionist models often offer good performance in pattern recognition and generalization, and present such qualities as natural learning ability, noise tolerance and graceful degradation. By contrast, symbolic models often present a complementary profile: they offer good performance in reasoning and deduction, and present such qualities as natural symbolic manipulation and explanation abilities. In the context of this paper, we address two limitations of artificial neural networks: the lack of explicit knowledge and the absence of temporal aspect in their implementation. *STN* : is a model of a specialized temporal neuron which includes both symbolic and temporal aspects. To illustrate the *STN* utility, we consider a system for phoneme recognition.

1 Introduction

The automatic speech recognition (ASR) is the process whereby the machine tries “to decode” the speech signal. Most of the current speech recognition systems are based on hidden Markov models (HMMs) techniques [2],[4]. Another approach besides HMM’s are the connectionist techniques[3],[6].

Artificial neural networks (ANNs) are good pattern recognisers, they are able to recognize patterns even when data are noisy, ambiguous or distorted [3]. Albeit, the problem in neural networks is with the choice of architecture (the only way to decide on a certain architecture is on a trial-and-error basis) and the lack of explanation. Researches in this area deal with the integration of symbolic knowledge insight of the connectionist architecture [5],[7].

The purpose of this paper, is to introduce a symbolic neural network dedicated to the speech recognition. The particularity of the proposed system with respect to those available in literature is the introduction of both the temporal and the symbolic aspects.

The remainder of the paper is structured as follows: The second section, defines the automatic speech recognition, then we introduce the ANNs. Section 3, gives an overview of the whole project, which is dedicated to speech recognition. In section 4, we describe the conceptual elements of the first layer. Section 5, describes the decision layer and particularly the *STN* model. In section 6, practical issues of the application are described for phonemes recognition. Finally, a conclusion is drawn.

2 Speech Recognition and Artificial Neural Networks

2.1 Speech Recognition

The speech recognition task involves several stages, the most important of them is the features extraction stage. In this stage, a smallest set of features will represent the original signal. Then the obtained representation of the signal is compared to the reference patterns to determine the closest one.

Most of the methods used in speech recognition (and in pattern recognition) include two stages : the training and the recognition stages. In the training stage, we present to the system a set of examples and at the end of the stage, the system will be able to distinguish them correctly, in this stage, patterns which were not in the training set are presented to the system, and it should categorize them correctly.

2.2 Artificial Neural Networks

Artificial neural networks are systems composed of a large number of simple interconnected units that simulate brain activity. Each of these units, that are the equivalent of the neuron in a biological simulation, is a part of layered structure, and produces an output that is a non-linear function of the inputs. In the feed forward networks such as the multilayer perceptron (MLP), the output of each layer of units is connected

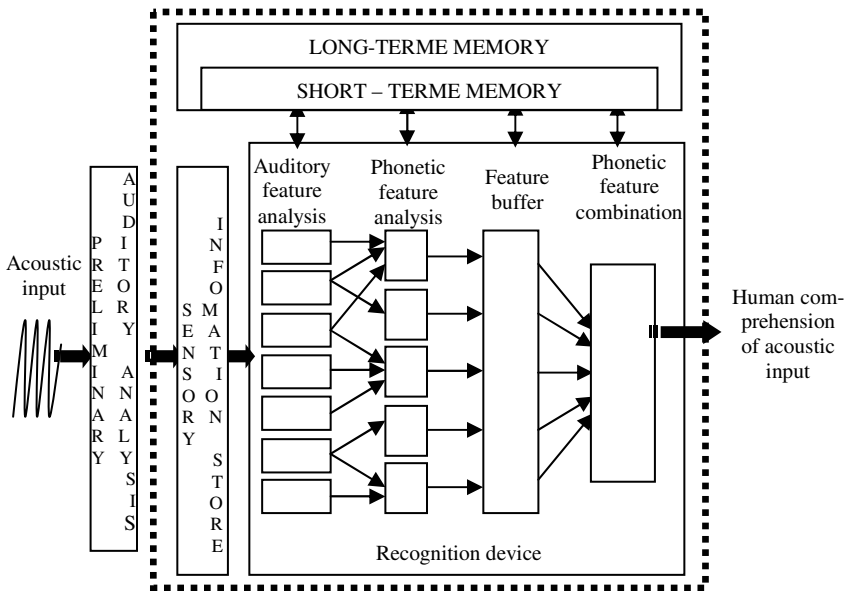


Fig. 1. Conceptual block diagram of human speech understanding (after [4])

to units of a higher layer with directed connections that are the equivalent to brain synapses. In the MLP The first layer is called the input layer, the last one is the output layer, and between there may be one or more hidden layers.

2.3 ANNs and Speech Recognition

Figure 1 shows a conceptual block diagram of a speech understanding system loosely based on a model of speech perception in human beings [4]. This diagram clearly underlines the importance of connectionist models when modelling such applications. Early attempts to model speech recognizers uses classical MLP, these approaches assume static representation of the time, later attempts were made to consider the dynamic aspect, the most popular of them is the TDANN (Time Delay Artificial Neural Networks) introduced by Waibel (see [6]), here the time is not explicitly represented in the network and the structure is too much complicated. A connectionist expert system dedicated to speech recognition was presented in [1], although, this system did not consider explicitly the temporal parameter.

3 *NESSR*: Neural Expert System for Speech Recognition

3.1 The System Overview

The overall system comprises three components : a recognition memory, a short term memory and a long term memory. *NESSR* is the recognition memory, which is a neural expert network. *NESSR* is a modular network, the first module is concerned with the phoneme recognition, the second one recognizes the words. The short-term memory is the memory where temporary events are stored, which may occur during the inferencing process (see § 5. 3). The long-term memory is the memory where are stored high level information of the language, this will validate a given decision. The role of this memory is beyond the scope of this paper.

3.2 Integrating Symbols Insight of the Network [1]

We consider an MLP, so neurons are regrouped into layers which correspond to the levels of our application which consists on the isolated layer word recognition. Thus, the input layer represents the acoustical level, the hidden layer the phonetic level, and the

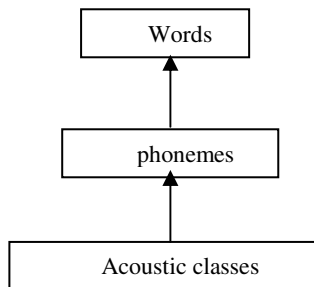


Fig. 2. *NESSR* topology

output layer, stands for the lexical one (figure 2). For the purpose of this paper, we are only concerned with the two first levels. So, the considered objects are : phonemes and their acoustic characteristics.

4 The Sensory Layer: The Acoustic Level

The input layer detects changes in the environment. Neurons of the sensory layer captured particularities of the pattern in entry of the network. In the speech recognition context, the cells detect features of the signal.

4.1 The Neurons Structure: Specialized Neurons

Since these cells belong to a symbolic network; every cell is specialized in the detection of one characteristic of the signal. We consider these characteristics as acoustic classes, so, we call a neuron of this layer : neuron-class. These particularities did not have a particular physical significance, they are numbered from 1 to n.

4.2 How to Determine Acoustic Classes ?

The first stage of the ASR process provides a collection of numeric vectors from the digitised signal. To translate this representation to a symbolic space, we perform a vector quantization (VQ) over all available vectors in the training stage. VQ enables us to replace each acoustic vector by the correspondent discreet symbol, where symbols represent entries of the code-book. So, there are as many neuron-class as there are entries in the codebook.

4.3 Activation of a Neuron

A neuron-class fires if the associated characteristic is detected in the signal. The network dynamic is triggered by discreet instants. At a given instant t , only one neuron-class is active. This supposes that the presentation of a signal to the network lasts from the instant t_0 to the instant t_n . In this interval of time many neurons can be activated.

We notice that the successive activation of the same neuron is taken into account by the network (see the *STN* model properties).

5 The Decision Layer: The Phonetic Level

Activations of the sensory layer are transmitted to the following layer whose role is to associate to the acoustic entries a phonetic units of the language; in this case phonemes. To a detected sequence of acoustic classes will be associated one phoneme. The recognition of a phoneme leads to an implicit segmentation of the signal at this point of the structure.

5.1 Structure of Neurons: Specialized Temporal Neurons

As for the sensory neurons the cells of this layer are meaningful. In this case every cell represents one phoneme of the Arabic language, we will call it : neuron-phoneme.

A phoneme is defined by the detection of a sequence of acoustic classes. Thus, when there is correlation between the detection of an acoustic class and the recognition of a phoneme, a connection between the concerned neurons is initiated. The activation of these entries must be in a very definite order assured by the structure of the neuron, in which an entry i cannot be considered while the entry $i-1$ is not already pre-activated. To model such neuron we suggest the following neuron model (figure 3).

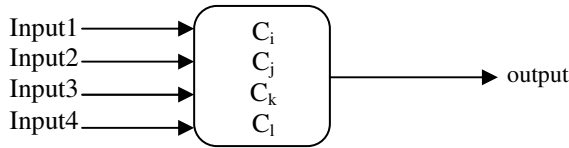


Fig. 3. The STN model

5.2 The Phoneme Characterization

To determine the needed acoustic classes for the detection of a phoneme : we consider the set of classes obtained after the VQ stage, and we operate a study of correlation between these prototypes and the set of phonemes. This permits us to extract the necessary set of classes for the recognition of any phoneme.

Table 1. Line of the correlation matrix

	C1	C2	C3	C4	C5	C6	C7	...	C64
a1_1	×				×	×			

×: detected class

Below is a line of a table illustrating relations between the definite classes and the apparition of a phoneme. This table is automatically built ; each occurrence of a phoneme from the training set is analysed, then quantified. When a characteristic appears in the signal, a mark is set in corresponding phoneme box. If a class appears more than 90% in a phoneme, we consider that it is basic constituent of the phoneme. Once characteristics of the phoneme are designated their order is established.

5.3 Activation of the STN

When a characteristic C_i is detected the associated neuron-class fires and all connections from this neuron are pre-activated. All neuron-phonemes whose first characteristic is C_i are pre-activated. Thus, a neuron-phoneme is pre-activated as soon as its first entry is pre-active, this supposes that several neuron-phonemes can be simultaneously pre-activated. Albeit, a neuron-phoneme fires only if all its entries are activated. When a neuron-phoneme fires all connections coming from the previous layer are deactivated, it is the same way for all competitor neurons, i.e. those which were simultaneously pre-active. If the detection of a characteristic can provoke the activation of more than one target cell, only one cell fires and this information is stored in the short-term memory. Let's notice that this situation is very rare (considering the number of classes 64).

5.4 Illustrative Example

In the figure 8, we present an example to illustrate the particular situations that constitute limit conditions of the model, and justify its use in temporal applications.

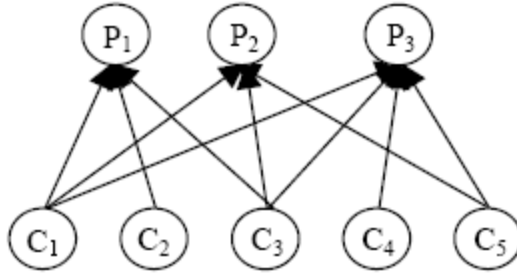


Fig. 4. Network connections to illustrate the *STN* activations

Table 2. Activation example of neuron-phonemes

	Neurone-p1	Neurone-p2	Neurone-p3
T=0	—————> C1> C2> C3	—————> C1> C3> C5	—————> C1> C5> C4> C3
T=1	—————> C1> C2> C3	—————> C1> C3> C5	—————> C1> C5> C4> C3
T=2	—————> C1> C2> C3	—————> C1> C3> C5	—————> C1 —————> C5> C4> C3
T=3	—————> C1 —————> C2> C3	—————> C1> C3> C5	—————> C1 —————> C5> C4> C3
T=4	—————> C1 —————> C2 —————> C3	—————> C1 —————> C3> C5	—————> C1 —————> C5> C4> C3

.....> Non active Connection
 —————> Pre-active Connection

We consider the above network, and we assume the following sequence: $\dots C_1 C_1 C_5 C_2 C_3 C_4 \dots$, the activation of the network is transcribed in the following table.

At the instant $t = 0$, the characteristic C_1 is detected so the correspondent neuron-class fires, and all its output links are pre-activated ; this implies the pre-activation of the three neuron-phonemes of the network; because C_1 corresponds to the first entry for all these target neurons.

At the instant $t = 1$, the same characteristic is detected, but this second activation doesn't bring any change in the state of the network.

At the instant $t = 2$, the characteristic C_5 is detected, this induces the pre-activation of the entry C_5 of the neuron-p3. The C_5 entry of neuron-p2 could not be pre-activated, because it could not be considered before the connection C_3 is pre-activated.

At the instant $t = 3$, the characteristic C_2 is detected this pre-activates the second entry of neuron-p1.

At the instant $t = 4$, the characteristic C_3 is detected this pre-activates the corresponding entries in the target neurons p1 and p2. In this last pre-activation the neuron-p1 has its entries pre-active. At this moment, it fires and all connections as well as the other target neurons are deactivated.

After C_3 the sequence in entry of the network is segmented and the activation of neuron - p1 is propagated to the following layer. A new session of phoneme recognition starts with C_4 .

5.5 Particularities of the STN Model

The structure of the STN neuron we suggest to model phonemes, allows the successive detection of the same acoustic characteristic of the signal (C_1 in the previous example); i.e. the model allows stationary transitions of the signal. This structure also allows the insertion of less important classes in the phoneme among pertinent classes (in the previous example C_5 is inserted in p1 structure).

6 Experimental Results

To evaluate performances of this module of *NESSR*, we perform some experimentations related to Arabic phoneme recognition. In the following, we describe the practical stages:

6.1 Features Extraction

The context of the present work is phoneme recognition. For developing experimental results, a set of Arabic words, including all the phonemes, was used. This corpus comprises twenty five words. Words are segmented and labelled into phonemes. Words are recorded with pause between them, and are uttered by many people of the laboratory. All examples were uttered in relatively quiet room. The incoming signal is sampled at 11025 Hz, with 8 bits of precision, and the sampled signal is processed by a first-order digital filter in order to spectrally flatten the signal. Sections of 400

consecutive samples are blocked into a single frame, corresponding to $400/11.025 \approx 36$ ms. Frames are spaced M samples ($M=100$). Then the frames are individually multiplied by a N -sample window. In ASR the most-used window shape is the Hamming window. From each frame, we extract a set of 13 Mel Frequency Cepstral Coefficients (MFCCs).

6.2 Vector Quantization

We consider all acoustical vectors we obtain during the training stage, we regroup them into disjoint classes (64) using the k -means algorithm. At the recognition phase, the vector quantizer compares each acoustical vector v_j of the signal to stored vectors c_i (code-words), and v_j is coded by the vector c_b that best represents v_j according to some distortion measure d . $d(v_j, c_b) = \min(d(v_j, c_i))$, we use the Euclidian distance.

6.3 Results

The database comprises utterances of 25 words, uttered by 14 speakers, 8 of them participate in the training stage (when, we define the acoustic classes and the phoneme characteristics). To perform evaluation tests we form two groups : The group TS1, includes new utterances of speakers who have participate in the training stage. The group TS2, includes utterances of speakers who did not participate in the training stage and some of those who participate. In the table bellow, we mention results, we have obtained for the considered phonemes (phonemes are given in IPA notation ; /a/, /u/ and /i/ are Arabic vowels).

Table 3. recognition rate in %

	/a/	/u/	/i/	/m/	/H/
TS1	99.2	98.4	98.2	97.6	95.1
TS2	97.7	97	97	95.8	95

7 Conclusion

In this paper we have attempted to present our contribution in the separate fields of the neurosymbolic systems and the temporal connectionist models. Our suggestion tries to combine in the same network the two components throughout the proposition of a new neuron structure : we called *STN* model. An application of this model is proposed in the phoneme recognition.

Although the obtained recognition rates are under our hope they still being promising ones, and we still believe that the neural expert models are a promising trend in resolution of perception problems, since this category of problems involve both neural models and symbolic reasoning.

References

1. Bahi H., Sellami M., Système expert connexionniste pour la reconnaissance de la parole, proceedings of RFIA, Vol 2, pp: 659-665. Toulouse, France, 2004.
2. Becchitti C., Ricotti L. P., speech recognition: *theory and C++ implementation*, John Wiley, England, 1999.
3. Bishop C. M., Neural networks for pattern recognition , Clarendon Press, Oxford, 1995.
4. Rabiner L., Hwang B., Fundamentals of speech recognition, Prentice Hall, 1993.
5. Sun R., Alexandre F., Connectionist-Symbolic Integration: From Unified to Hybrid Approaches, Lawrence Erlbaum Associates, 1997.
6. Tebelski J., Speech recognition using neural networks, PhD Thesis, Carnegie Mellon University, May 1995.
7. Towell G., Symbolic knowledge and neural networks: Insertion, Refinement and extraction, PhD thesis, University of Wisconsin, Madison, 1991.

An Evidential Reasoning Approach to Weighted Combination of Classifiers for Word Sense Disambiguation

Cuong Anh Le¹, Van-Nam Huynh², and Akira Shimazu¹

¹ School of Information Science

² School of Knowledge Science, Japan Advanced Institute of Science and Technology,
1-1, Asahidai, Nomi, Ishikawa 923-1292 Japan
{cuonganh, huynh, shimazu}@jaist.ac.jp

Abstract. Arguing that various ways of using context in word sense disambiguation (WSD) can be considered as distinct representations of a polysemous word, a theoretical framework for the weighted combination of soft decisions generated by experts employing these distinct representations is proposed in this paper. Essentially, this approach is based on the Dempster-Shafer theory of evidence. By taking the confidence of individual classifiers into account, a general rule of weighted combination for classifiers is formulated, and then two particular combination schemes are derived. These proposed strategies are experimentally tested on the datasets for four polysemous words, namely *interest*, *line*, *serve*, and *hard*.

Keywords: Computational linguistics, Weighted combination of classifiers, Word sense disambiguation, Dempster-Shafer theory of evidence.

1 Introduction

Word sense disambiguation is a computational linguistics task recognized since the 1950s. Roughly speaking, word sense disambiguation involves the association of a given word in a text or discourse with a particular sense among numerous potential senses of that word. As mentioned in [5], this is an “intermediate task” necessarily to accomplish most natural language processing tasks. It is obviously essential for language understanding applications, while also at least helpful for other applications whose aim is not language understanding such as machine translation, information retrieval, among others. Since its inception, many methods involving WSD have been developed in the literature (see, e.g., [5] for a survey). During the last decade, many supervised machine learning algorithms have been used for this task, including Naïve Bayesian (NB) model, decision trees, exemplar-based model, support vector machine, maximum entropy, etc. As observed in studies of machine learning systems, although one could choose one of learning systems available to achieve the best performance for a given pattern recognition problem, the set of patterns misclassified by the different classification systems would not necessarily overlap. This means that

different classifiers may potentially offer complementary information about the patterns to be classified. This observation highly motivated the interest in combining classifiers during the recent years. Especially, classifier combination for WSD has unsurprisingly received much attention recently from the community as well, e.g., [6, 4, 12, 8, 3, 15].

As is well-known, there are basically two classifier combination scenarios. In the first scenario, all classifiers use the same representation of the input pattern. In the context of WSD, the work by Kilgarriff and Rosenzweig [6], Klein et al. [8], and Florian and Yarowsky [3] could be grouped into this first scenario. In the second scenario, each classifier uses its own representation of the input pattern. An important application of combining classifiers in this scenario is the possibility to integrate physically different types of features. In this sense, the work by Pedersen [12], Wang and Matsumoto [15] can be considered as belonging to this scenario. In this paper, we focus on the weighted combination of classifiers for WSD in the second scenario of combination strategies. Particularly, we first consider various ways of using context in WSD as distinct representations of a polysemous word under consideration, then all these representations are used as providing individual information sources to identify the meaning of the target word. We then develop a general framework for the weighted combination of individual classifiers corresponding to distinct representations. Essentially, this approach is based on Dempster-Shafer (DS) theory of evidence [13], which has been recently increasingly applied to classification problems, e.g. [2, 16]. Moreover, two combination strategies are developed and experimentally tested on the datasets for four polysemous words, namely *interest*, *line*, *serve*, and *hard*, and compared with previous studies.

The paper is organized as follows. In the next section, basic notions of DS theory will be briefly recalled. Section 3 reformulate the WSD problem so that the general framework for the weighted combination of classifiers can be formulated, and the two combination strategies can be developed. Section 4 discuss about context representation of a target word and presents our selection. Next, section 5 presents experimented results and the comparison with previous known results on the same test datasets. Finally, some conclusions are presented in Section 6.

2 Dempster-Shafer Theory of Evidence

In DS theory, a problem domain is represented by a finite set Θ of mutually exclusive and exhaustive hypotheses, called *frame of discernment* [13]. In the standard probability framework, all elements in Θ are assigned a probability. And when the degree of support for an event is known, the remainder of the support is automatically assigned to the negation of the event. On the other hand, in DS theory mass assignments are carried out for events as they know, and committing support for an event does not necessarily imply that the remaining support is committed to its negation. Formally, a basic probability assignment (BPA, for short) is a function $m : 2^\Theta \rightarrow [0, 1]$ verifying

$$m(\emptyset) = 0, \text{ and } \sum_{A \in 2^\Theta} m(A) = 1$$

The quantity $m(A)$ can be interpreted as a measure of the belief that is committed exactly to A , given the available evidence. A subset $A \in 2^\Theta$ with $m(A) > 0$ is called a *focal element* of m . A BPA m is called to be *vacuous* if $m(\Theta) = 1$ and $m(A) = 0$ for all $A \neq \Theta$.

Two evidential functions derived from the basic probability assignment m are the belief function Bel_m and the plausibility function Pl_m , defined as

$$Bel_m(A) = \sum_{\emptyset \neq B \subseteq A} m(B), \text{ and } Pl_m(A) = \sum_{B \cap A \neq \emptyset} m(B)$$

Two useful operations that play a central role in the manipulation of belief functions are *discounting* and *Dempster's rule of combination* [13]. The discounting operation is used when a source of information provides a BPA m , but one knows that this source has probability α of reliable. Then one may adopt $(1 - \alpha)$ as one's *discount rate*, which results in a new BPA m^α defined by

$$m^\alpha(A) = \alpha m(A), \text{ for any } A \subset \Theta \quad (1)$$

$$m^\alpha(\Theta) = (1 - \alpha) + \alpha m(\Theta) \quad (2)$$

Consider now two pieces of evidence on the same frame Θ represented by two BPAs m_1 and m_2 . Dempster's rule of combination is then used to generate a new BPA, denoted by $(m_1 \oplus m_2)$ (also called the orthogonal sum of m_1 and m_2), defined as follows.

$$\begin{aligned} (m_1 \oplus m_2)(\emptyset) &= 0, \\ (m_1 \oplus m_2)(A) &= \frac{1}{1 - \sum_{B \cap C = \emptyset} m_1(B)m_2(C)} \sum_{B \cap C = A} m_1(B)m_2(C) \end{aligned} \quad (3)$$

It is worth noting that Dempster rule of combination has some attractive features such as: it is commutative and associative; given two BPAs m_1 and m_2 , if m_1 is vacuous then $m_1 \oplus m_2 = m_2$.

3 Weighted Combination of Classifiers for WSD

In this section, after reformulating the WSD problem in terms of a pattern recognition problem with multi-representation of patterns. The general framework for weighted combination of classifiers is developed for WSD problem and then, two particular combination schemes are explored.

3.1 WSD with Multi-representation of Context

Given a polysemous word w , which may have M possible senses (classes): c_1, c_2, \dots, c_M , in a context C , the task is to determine the most appropriate sense of w . Generally, context C can be used in two ways [5]: in the *bag-of-words*

approach, the context is considered as words in some window surrounding the target word w ; in the *relational information based approach*, the context is considered in terms of some relation to the target such as distance from the target, syntactic relations, selectional preferences, phrasal collocation, semantic categories, etc. As such, for a target word w , we may have different representations of context C corresponding to different views of context. Assume we have such R representations of C , say $\mathbf{f}_1, \dots, \mathbf{f}_R$, serving for the aim of identifying the right sense of the target w .

Now let us assume that we have R classifiers, each representing the context by a distinct set of features. The set of features \mathbf{f}_i , which is considered as a representation of context C of the target w , is used by the i -th classifier. Furthermore, assume that each i -th classifier (expert) is associated with a weight α_i , $0 \leq \alpha_i \leq 1$, reflecting the relative confidence in it, which may be interpreted as reliable probability of the i -th classifier in its prediction. As such representations \mathbf{f}_i 's ($i = 1, \dots, R$) are considered as distinct information sources associated with corresponding weights serving for identifying the sense of the target w . The problem now is how to combine these information sources to reach a consensus decision for identifying the sense of w .

3.2 A General Framework

Given a target word w in a context C and $\mathcal{S} = \{c_1, c_2, \dots, c_M\}$ is the set of its possible senses. Using the vocabulary of DS theory, \mathcal{S} can be called the *frame of discernment* of the problem. As mentioned above, various ways of using the context could be considered as providing different information sources to identify the meaning of the target word. Each of these information sources does not by itself provide 100% certainty as a whole piece of evidence for identifying the sense of the target. Formally, we have the available information for making the final decision on the sense of w given as follows

- R probability distributions $P(\cdot|\mathbf{f}_i)$ ($i = 1, \dots, R$) on \mathcal{S} ,
- the weights α_i of the individual information sources ($i = 1, \dots, R$)¹.

From the probabilistic point of view, we may straightforwardly think of the combiner as a weighted mixture of individual classifiers defined as

$$P(c_k|\mathbf{f}_1, \dots, \mathbf{f}_R) = \frac{1}{\sum_i \alpha_i} \sum_{i=1}^R \alpha_i P(c_k|\mathbf{f}_i), \text{ for } k = 1, \dots, M \tag{4}$$

Then the target word w should be naturally assigned to the sense c_j according to the following decision rule

$$j = \arg \max_k P(c_k|\mathbf{f}_1, \dots, \mathbf{f}_R) \tag{5}$$

However, by considering the problem as that of weighted combination of evidence for decision making, in the following we will formulate a general rule

¹ Note that the constraint $\sum_i \alpha_i = 1$ does not need to be imposed.

of combination based on DS theory. To this end, we first adopt a probabilistic interpretation of weights. That is, the weight α_i ($i = 1, \dots, R$) is interpreted as reliable probability of the i -th classifier. This interpretation of weights seems to be especially appropriate when defining weights in terms of the accuracy of individual classifiers.

Under such an interpretation of weights, the piece of evidence represented by $P(\cdot|\mathbf{f}_i)$ should be discounted at a discount rate of $(1 - \alpha_i)$. This results in a BPA m_i verifying

$$m_i(\{c_k\}) = \alpha_i P(c_k|\mathbf{f}_i) \triangleq p_{i,k}, \text{ for } k = 1, \dots, M \quad (6)$$

$$m_i(\mathcal{S}) = 1 - \alpha_i \triangleq p_{i,\mathcal{S}} \quad (7)$$

$$m_i(A) = 0, \forall A \in 2^{\mathcal{S}} \setminus \{\mathcal{S}, \{c_1\}, \dots, \{c_M\}\} \quad (8)$$

That is, the discount rate of $(1 - \alpha_i)$ can not be distributed to anything else than \mathcal{S} , the whole frame of discernment. We are now ready to formulate our belief on the decision problem by aggregating all pieces of evidence represented by m_i 's in the general form of the following

$$m = \bigoplus_{i=1}^R m_i \quad (9)$$

where m is a BPA and \oplus is a combination operator in general.

3.3 The Discounting-and-Orthogonal Sum Combination Strategy

As discussed above, we consider each $P(\cdot|\mathbf{f}_i)$ as the belief quantified from the information source \mathbf{f}_i and the weight α_i as a ‘‘degree of trust’’ of \mathbf{f}_i supporting the identification for the sense of w as a whole. As mentioned in [13], an obvious way to use discounting with Dempster’s rule of combination is to discount all BPAs $P(\cdot|\mathbf{f}_i)$ ($i = 1, \dots, R$) at corresponding rates $(1 - \alpha_i)$ ($i = 1, \dots, R$) before combining them. Thus, Dempster’s rule of combination now allows us to combine BPAs m_i ($i = 1, \dots, R$) under the independent assumption of information sources for generating the BPA m , i.e. \oplus in (9) is the orthogonal sum operation.

Note that, by definition, focal elements of each m_i are either singleton sets or the whole set \mathcal{S} . It is easy to see that m also verifies this property if applicable. Interestingly, the commutative and associative properties of the orthogonal sum operation with respect to a combinable collection of BPAs m_i ($i = 1, \dots, M$) and the mentioned property essentially form the basis for developing a recursive algorithm for calculation of the BPA m . This can be done as follows.

Let $I(i) = \{1, \dots, i\}$ be the subset consisting of first i indexes of the set $\{1, \dots, R\}$. Assume that $m_{I(i)}$ is the result of combining the first i BPAs m_j , for $j = 1, \dots, i$. Let us denote

$$p_{I(i),k} \triangleq m_{I(i)}(\{c_k\}), \text{ for } k = 1, \dots, M \quad (10)$$

$$p_{I(i),\mathcal{S}} \triangleq m_{I(i)}(\mathcal{S}) \quad (11)$$

With these notations and (6)–(7), the key step in the combination algorithm is to inductively calculate $p_{I(i+1),k}$ ($k = 1, \dots, M$) and $p_{I(i+1),\mathcal{S}}$ as follows

$$p_{I(i+1),k} = \frac{1}{\kappa_{I(i+1)}} [p_{I(i),k}p_{i+1,k} + p_{I(i),k}p_{i+1,\mathcal{S}} + p_{I(i),\mathcal{S}}p_{i+1,k}] \tag{12}$$

$$p_{I(i+1),\mathcal{S}} = \frac{1}{\kappa_{I(i+1)}} (p_{I(i),\mathcal{S}}p_{i+1,\mathcal{S}}) \tag{13}$$

for $k = 1, \dots, M$, $i = 1, \dots, R - 1$, and $\kappa_{I(i+1)}$ is a normalizing factor defined by

$$\kappa_{I(i+1)} = \left[1 - \sum_{j=1}^M \sum_{\substack{k=1 \\ k \neq j}}^M p_{I(i),j}p_{i+1,k} \right] \tag{14}$$

Finally, we obtain m as $m_{I(R)}$. For the purpose of decision making, we now define a probability function P_m on \mathcal{S} derived from m via the *pignistic transformation* as follows

$$P_m(c_k) = m(\{c_k\}) + \frac{1}{M}m(\mathcal{S}) \text{ for } k = 1, \dots, M \tag{15}$$

and we have the following decision rule:

$$j = \arg \max_k P_m(c_k) \tag{16}$$

It would be interesting to note that an issue may arise with the orthogonal sum operation, that is the use of the total probability mass κ associated with conflict as defined in the normalization factor. Consequently, applying it in an aggregation process may yield counterintuitive results in the face of significant conflict in certain situations as pointed out in [17]. Fortunately, in the context of the weighted combination of classifiers, by discounting all $P(\cdot|\mathbf{f}_i)$ ($i = 1, \dots, R$) at corresponding rates $(1 - \alpha_i)$ ($i = 1, \dots, R$), we actually reduce conflict between the individual classifiers before combining them.

3.4 The Discounting-and-Averaging Combination Strategy

In this strategy, instead of using Dempster’s rule of combination after discounting $P(\cdot|\mathbf{f}_i)$ at the discount rate of $(1 - \alpha_i)$, we apply the averaging operation over BPAs m_i ($i = 1, \dots, R$) to obtain the BPA m defined by

$$m(A) = \frac{1}{R} \sum_{i=1}^R m_i(A) \tag{17}$$

for any $A \in 2^{\mathcal{S}}$. By definition, we get

$$m(\{c_k\}) = \frac{1}{R} \sum_{i=1}^R \alpha_i P(c_k|\mathbf{f}_i), \text{ for } k = 1, \dots, M \tag{18}$$

$$m(\mathcal{S}) = 1 - \frac{\sum_{i=1}^R \alpha_i}{R} \triangleq 1 - \bar{\alpha} \tag{19}$$

$$m(A) = 0, \forall A \in 2^{\mathcal{S}} \setminus \{\mathcal{S}, \{c_1\}, \dots, \{c_M\}\} \tag{20}$$

Note that the probability mass is unassigned to individual classes but the whole frame of discernment \mathcal{S} , $m(\mathcal{S})$, is the average of discount rates. Therefore, instead of allocating the average discount rate $(1 - \bar{\alpha})$ to $m(\mathcal{S})$ as above, we use it as a normalization factor and easily obtain

$$m(\{c_k\}) = \frac{1}{\sum_i \alpha_i} \sum_{i=1}^R \alpha_i P(c_k | \mathbf{f}_i), \text{ for } k = 1, \dots, M \quad (21)$$

$$m(A) = 0, \forall A \in 2^{\mathcal{S}} \setminus \{\{c_1\}, \dots, \{c_M\}\} \quad (22)$$

which interestingly turns out to be the weighted mixture of individual classifiers as defined in (4). Then we have the decision rule (5).

4 Representations of Context for WSD

Context plays an essentially important role in WSD and the representation choice of context is a factor which may be more important than the algorithm used for the task itself on the aspect of affecting the obtained result. For predicting senses of a word, information usually used in all studies is the topic context which is represented by bag of words. Ng and Lee [11] proposed the use of more linguistic knowledge resources including topic context, collocation of words, and a syntactic relationship verb-object, which then became popular resources for determining word sense in many papers. In [9], the authors use another information type, which is words or part-of-speech and each is assigned with its position in relation with the target word. However, in the second scenario of classifier combination strategies, according to our knowledge, only topic context with different sizes of context windows is used for creating different representations of a polysemous word, such as in Pedersen [12] and Wang and Matsumoto [15].

On the other hand, we observe that two of the most important information sources for determining the sense of a polysemous word are the topic of context and relational information representing the structural relations between the target word and the surrounding words in a local context. Under such an observation, we have experimentally designed five kinds of representation defined as follows: \mathbf{f}_1 is a set of unordered words in the large context; \mathbf{f}_2 is a set of words assigned with their positions in the local context; \mathbf{f}_3 is a set of part-of-speech tags assigned with their positions in the local context; \mathbf{f}_4 is a set of collocations of words; \mathbf{f}_5 is a set of collocations of part-of-speech tags. Symbolically, we have

- $\mathbf{f}_1 = \{w_{-n_1}, \dots, w_{-2}, w_{-1}, w_1, w_2, \dots, w_{n_1}\}$
- $\mathbf{f}_2 = \{(w_{-n_2}, -n_2), \dots, (w_{-1}, -1), (w_1, 1), \dots, (w_{n_2}, n_2)\}$
- $\mathbf{f}_3 = \{(p_{-n_3}, -n_3), \dots, (p_{-2}, -2), (p_{-1}, -1), (p_1, 1), (p_2, 2), \dots, (p_{n_3}, n_3)\}$
- $\mathbf{f}_4 = \{w_{-l} \cdots w_{-1} w w_1 \cdots w_r \mid l + r \leq n_4\}$
- $\mathbf{f}_5 = \{p_{-l} \cdots p_{-1} w p_1 \cdots p_r \mid l + r \leq n_5\}$

where w_i is the word at position i in the context of the ambiguous word w and p_i be the part-of-speech tag of w_i , with the convention that the target word

w appears precisely at position 0 and i will be negative (positive) if w_i appears on the left (right) of w . In the experiment, we design the window size of topic context (for both left and right windows) as 50 for the representation \mathbf{f}_1 , i.e. $n_1 = 50$, while the window size n_i of local context as 3 for remaining representations.

5 Experiments

5.1 Computing the Probabilities and Determining Weights

In the experiment, each individual classifier is a naive Bayesian classifier built on a context representation. We have five individual classifiers corresponding to five context representations as mentioned above. As we have seen above, in the weighted combination of classifiers we need to compute the a posteriori probabilities $P(c_k|\mathbf{f}_i)$. For the context C , suppose that the representation \mathbf{f}_i of C is represented by a set of features $\mathbf{f}_i = (f_{i,1}, f_{i,2}, \dots, f_{i,n_i})$ with the assumption that the features $f_{i,j}$ are conditionally independent, and then $P(c_k|\mathbf{f}_i)$ is computed using the following formula based on Bayes theorem.

$$P(c_k|\mathbf{f}_i) = \frac{P(\mathbf{f}_i|c_k)P(c_k)}{P(\mathbf{f}_i)} = \frac{P(c_k) \prod_{j=1}^{n_i} P(f_{i,j}|c_k)}{P(\mathbf{f}_i)} \tag{23}$$

In the experiment, we used 10-fold cross validation on the training data and then the obtained accuracies of the individual classifiers are used for weights α_i . Although we determine the weights based on the accuracies of individual classifiers, other methods of identifying the weights α_i such as using linear regression and least-squares-fit could be used. However, this is left for the long version of this paper.

5.2 Data and Result

We tested on the datasets of four words, namely *interest*, *line*, *serve*, and *hard*, which are used in numerous comparative studies of word sense disambiguation methodologies such as Pedersen [12], Ng and Lee [11], Bruce & Wiebe [1], and Leacock and Chodorow [9]. There are 2369 instances of *interest* with 6 senses, 4143 instances of *line* with 6 senses, 4378 instances of *serve* with 4 senses, and 4342 instances of *hard* with 3 senses.

In the experiment, we obtained the results using 10-fold cross validation. Table 1 shows the results obtained by using two strategies of weighted combination of classifiers and the best results obtained by individual classifiers respectively. It is shown that both combination strategies give better results than the best individual classifier in all cases. Interestingly also, the results showed that in all cases the orthogonal sum based combination strategy is better than that based on weighted sum. This can be experimentally interpreted as follows. In our multi-representation of context, each individual classifier corresponds to a

Table 1. Results using the proposed methods and some results from previous studies. In the table, BW, M, NL, LC, and P respectively abbreviate for Bruce & Wiebe [1], Mooney [10], Ng & Lee [11], Leacock & Chodorow [9], and Pedersen [12]

(%)	BW	M	NL	LC	P	The proposed method		
						best individual classifier	based on weighted sum	based on orthogonal sum
<i>interest</i>	78	–	87	–	89	86.8	90.7	90.9
<i>line</i>	–	72	–	84	88	82.8	85.6	87.2
<i>hard</i>	–	–	–	83	–	90.2	91	91.5
<i>serve</i>	–	–	–	83	–	84.4	89	89.7

type of features so that the conditional independence assumption seems to be realistic and, consequently, the orthogonal sum based combination strategy is a suitable choice for this scheme of multi-representation of context. In addition, Table 1 also shows that both combination strategies also give better results than previous work in all cases, with the exception of *line* which corresponds to Pedersen’s method as the best.

6 Conclusion

In this paper we first argued that various ways of using context in WSD can be considered as distinct representations of a polysemous word under consideration, then these representations assigned with weights are jointed into an account to identify the meaning of the target word. Based on DS theory of evidence, we developed a general framework for the weighted combination of individual classifiers corresponding to distinct representations. Moreover, two combination strategies have been developed and experimentally tested on the datasets for four polysemous words, namely *interest*, *line*, *serve*, and *hard*, and compared with previous studies. It has been shown that considering multi-representation of context significantly improves the accuracy of WSD by combining classifiers, as individual classifiers corresponding to different types of representation suitably offer complementary information about the target to be assigned to a sense. The experiment also shows that the combination strategy based on orthogonal sum is a suitable choice for this scheme of multi-representation of context.

Acknowledgement

This research is partly conducted as a program for the of “Fostering Talent in Emergent Research Fields” in Special Coordination Funds for Promoting Science and Technology by the Japanese Ministry of Education, Culture, Sports, Science and Technology.

References

1. Bruce, R. and Wiebe, J. 1994. Word-Sense Disambiguation using Decomposable Models. *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 139–145.
2. Denoeux, T., A k -nearest neighbor classification rule based on Dempster-Shafer theory, *IEEE Transactions on Systems, Man and Cybernetics* **25** (1995) 804–813.
3. Florian, R., and D. Yarowsky, Modeling consensus: Classifier combination for Word Sense Disambiguation, *Proceedings of EMNLP 2002*, pp. 25–32.
4. Hoste, V., I. Hendrickx, W. Daelemans, and A. van den Bosch, Parameter optimization for machine-learning of word sense disambiguation, *Natural Language Engineering* **8** (3) (2002) 311–325.
5. Ide, N., J. Véronis, Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art, *Computational Linguistics* **24** (1998) 1–40.
6. Kilgarriff, A., and J. Rosenzweig, Framework and results for English SENSEVAL, *Computers and the Humanities* **36** (2000) 15–48.
7. Kittler, J., M. Hatef, R. P. W. Duin, and J. Matas, On combining classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20** (3) (1998) 226–239.
8. Klein, D., K. Toutanova, H. Tolga Ilhan, S. D. Kamvar, and C. D. Manning, Combining heterogeneous classifiers for Word-Sense Disambiguation, *ACL WSD Workshop*, 2002, pp. 74–80.
9. Leacock, C., M. Chodorow, and G. Miller, Using corpus statistics and WordNet relations for Sense Identification, *Computational Linguistics* **24** (1998) 147–165.
10. Mooney, R. J., Comparative experiments on Disambiguating Word Senses: An illustration of the role of bias in machine learning, *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1996, pp. 82–91.
11. Ng, H. T., and H. B. Lee, Integrating multiple knowledge sources to Disambiguate Word Sense: An exemplar-based approach, *Proceedings of the 34th Annual Meeting of the Society for Computational Linguistics (ACL)*, 1996, pp. 40–47.
12. Pedersen, T., A simple approach to building ensembles of Naive Bayesian classifiers for Word Sense Disambiguation, *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2000, pp. 63–69.
13. Shafer, G., *A Mathematical Theory of Evidence* (Princeton University Press, Princeton, 1976).
14. Smets, P. and R. Kennes, The transferable belief model, *Artificial Intelligence* **66** (1994) 191–234.
15. Wang, X. J., and Y. Matsumoto, Trajectory based word sense disambiguation, *Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, August 2004, pp. 903–909.
16. Wang, H., and D. Bell, Extended k -nearest neighbours based on evidence theory, *The Computer Journal* **47** (6) (2004) 662–672.
17. Zadeh, L. A., Reviews of Books: A Mathematical Theory of Evidence, *The AI Magazine* **5** (1984) 81–83.

Signature-Based Approach for Intrusion Detection

Bon K. Sy

Queens College/CUNY, Computer Science Department,
Flushing NY 11367, U.S.A
bon@bunny.cs.qc.edu

Abstract. This research presents a data mining technique for discovering masquerader intrusion. User/system access data are used as a basis for deriving statistically significant event patterns. These patterns could be considered as a user/system access signature. Signature-based approach employs a model discovery technique to derive a reference ground model accounting for the user/system access data. A unique characteristic of this reference ground model is that it captures the statistical characteristics of the access signature, thus providing a basis for reasoning the existence of a security intrusion based on comparing real time access signature with that embedded in the reference ground model. The effectiveness of this approach will be evaluated based on comparative performance using a publicly available data set that contains user masquerade.

1 Introduction

Different kinds of security intrusion could occur in a networked computing environment [1]. For example, network intrusion could be launched via a denial of service attack, while system intrusion in the application layer (or layer 7) could occur through user masquerade. Intrusion prevention involves IT security professions to define security policy rules that can be translated into event patterns that, through real time monitoring, could trigger an alert for a potential intrusion [2, 3].

The challenge for intrusion detection is to develop scalable, extensible data mining techniques that can efficiently examine the audit trails in real time to accurately pinpoint the occurrence of an intrusion. Instead of relying on event patterns that attempt to capture an intrusion, we propose to rely on event patterns that attempt to capture what is expected to be the normal behavior of users and systems. In other words, our research is focused on developing models that signify the access signature as opposed to the intrusion signature. The rationale behind this shift in paradigm is that data are readily available to derive the statistical information about the event patterns, and thus the access signature. On the other hand, significant statistical information from sporadic intrusion activities may hardly be available, if any.

In this research we propose a signature-based approach for discovering masquerader intrusion. Masquerader intrusion refers to an intruder who executes system commands or requests system services under the identity of someone else,

often other legitimate user. In our proposed signature-based approach, statistically significant event patterns that characterize the user/system access behavior will be identified based on the concept of association patterns discussed in our previous research [9]. These statistically significant event patterns will be used to define a unique signature about the user/system access behavior. A probability model, referred to as a reference model, preserving the statistical information embedded in the unique signature will then be derived [10]. In the production cycle, statistically significant event patterns will be derived using windowed sequential real time user/system access data, and these event patterns of a windowed sequential data block defines a transitional signature. Inference about the existence of an intrusion will then be based on the degree of statistical deviation as measured by comparing the transitional signature with that embedded in the reference model.

2 Background Discussions

The signature based approach for intrusion detection presented in this paper could be considered as a behavior-based approach for statistical anomaly detection; where the essence of the signature based approach is to capture normal behavior — as opposed to unusual behavior — in terms of signature patterns.

Many different well-known techniques have been proposed for statistical anomaly detection. Many such techniques rely on detecting change point or outlier. One common approach towards change point or outlier detection is to determine how much an observed event (in question) is deviated from some reference “normal event set” using a distance measurement such as L-norm, Hamming distance, Manhattan distance, or vector cosine measure. Another common approach [13] is to determine whether an observed event (in question) appears in the low density regions of the probability distribution characterizing the “normal event set.” More recently, novel approach based on the use of n-gram matching rule [14] for positive and negative detection, as well as hybrid Markov model chain and rarity index model (based on extending the STIDE model) were also proposed [6,7].

In comparison to the existing techniques for statistical anomaly detection, signature based approach presented in this paper is unique in two regards. First, “normal event set” is characterized by a set of statistically significant association patterns referred to as a signature. These statistically significant association patterns bear an important information-theoretic characteristic; namely, frequently co-occurred events in a pattern do not just happen by chance as measured by mutual information criterion. Second, the distance measurement is then conducted under a two-way mutual comparison as opposed to a one-way comparison as typical in standard posterior probability measure. In a one-way comparison, observed event sample is compared against normal event observation. In a two-way comparison, access signature is compared against the observed model (of possible intrusion), and the signature of observed events (of possible intrusion) is compared against the access model to arrive a composite measurement. These are the distinctions of the signature-based approach in comparison to other approaches such as rarity criterion or posteriori probability based matching rule of n-gram samples.

3 Deriving Statistically Significant Patterns

In this research, the type of intrusion we focus on is masquerader intrusion in a Unix/Linux environment; i.e., an intruder injects operating system commands into the shell environment for a command execution under someone else identity. In a Unix/Linux system, “praudit” utility can be installed to keep track of the command execution history of a user [4]. Consider the following example of the command execution history of a user since a successful login session is established: pine, emacs, netscape, ssh, chmod, sftp, javac, java,

The problem of discovering masquerader intrusion is to determine from the command execution history such as the one shown above whether some command(s) in the command execution history is/are injected by an intruder but not issued by the user who owns the successful login session.

While it is conceivable to define sequential intrusion patterns as a basis for intrusion detection, there are three fundamental challenges of this approach [5,6,7]. First, the size of the security policy rule set and the corresponding intrusion patterns will grow over time as new intrusion methodologies are discovered. Second, real world intrusion seldom occurs frequent enough to accumulate statistical evidence for timely intrusion detection. Third, it may not be possible to always define security policy rules without causing conflict to what may be an expected acceptable activities. Consider a general security policy: “Change of the file access privilege on the password table should be trapped and interpreted as a potential intrusion,” this may be translated to an event trigger defined by “chmod 770 /etc/passwd”. Yet such a policy will cause interference on backup/recovery during the regular maintenance process.

To address the limitations just mentioned, we propose a signature concept that attempts to capture the unique characteristic of a legitimate user. The premise of applying the concept of signature is that there exist some unique access patterns of a legitimate user. Imagine in an extreme case where each legitimate user always performs the same activity upon establishing a successful login session; e.g., checking email (using pine), launching emacs to write a report to the supervisor, launching netscape to check company news events, ... etc. The likelihood of having two or more users with identical command execution sequence would be very small. Therefore, one may consider the entire command execution sequence of a user as an access signature. This is similar to the idea of *uniqueness* for intrusion detection discussed elsewhere [7]. Obviously defining an access signature based on the entire command execution sequence is unlikely to be computationally manageable. In addition, no user will have the execution sequence completely identical upon different successful login sessions — even certain commands or command sequence may always co-occur and appear as association patterns. An alternative approach is to consider categorizing the commands into few categories and to focus on low order association patterns [8] as an access signature. In doing so, it would be relatively more computationally manageable while we try to “optimize” the uniqueness of the access signature of a user.

In this research every Unix/Linux command is categorized into one of the following five groups: (1) Networking, (2) OS/System application/shell script, (3) File access, (4) Security, and (5) Communication.

The example of the command execution history shown earlier “pine, emacs, netscape, ssh, chmod, sftp, javac, java, ...” can then be translated into a category sequence “5 3 2 5 4 3 2 2...” Furthermore, the category sequence can be shifted and aligned when considering the low order association event patterns. In this example, shift and alignment for considering 4th order association patterns that accounts for the 4-tuple patterns (x1 x2 x3 x4) of the command execution history will be (x1:5 x2:3 x3:2 x4:5), (x1:3 x2:2 x3:5 x4:4), (x1:2 x2:5 x3:4 x4:3), ... etc. In this research, an access signature is defined as the collection of the statistically significant association patterns of 4th order (x1 x2 x3 x4) using the criteria below [9]:

$$\text{Support measure } Pr(x1, x2, x3, x4) \geq \text{some predefined threshold} \quad (1),$$

and

$$MI(x1, x2, x3, x4) \rightarrow \left(\frac{1}{Pr(x1, x2, x3, x4)} \right) \left(\frac{\chi^2}{2N} \right)^{\left(\frac{\hat{E}}{E'} \right)^{0/2}} \quad (2)$$

where $MI(x1, x2, x3, x4) = \text{Log}_2 Pr(x1 \ x2 \ x3 \ x4) / Pr(x1)Pr(x2)Pr(x3)Pr(x4)$

N = sample population size

χ^2 = Pearson chi-square test statistic defined as $(oi - ei)^2/ei$ with

oi = observed count = $N Pr(x1 \ x2 \ x3 \ x4)$

ei = expected count under the assumption of independence

\hat{E} = Expected entropy measure of estimated probability model

E' = Maximum possible entropy of estimated probability model

O = order of the association pattern (i.e., 4 in this case)

The choice of the 4th order association patterns is ad-hoc but under a careful consideration on balancing the representational and computational complexities. Further details about statistically significant patterns could be found in our previous paper [9]. Note that the above two criteria guarantee that any pattern considered statistically significant would have appeared frequently, and the co-occurrence of the associated events in a pattern does not just happen independently and by chance [10].

Since there are only 625 4th order association patterns for five command categories, one could argue that an intruder just has to run commands that belong to the same group as the legitimate user to reduce the chances of detection. This is true under the assumption that the intruder has the prior knowledge about the behavior of the legitimate user. If this is the case, no behavior-based intrusion detection will succeed because the intruder and legitimate user will no longer be distinguishable. And if the intruder is trying to guess the command sequence of the patterns that represent the access signature, there are $C(625, k)$ combinations; where k is the number of patterns defining the access signature. In this case, we will want to define the time period within which the legitimate user must reveal the access signature, while the likelihood of guessing the correct set of patterns defining the access signature is low.

4 Identifying Probability Reference Model

Referring to the example in the previous section, there are $5^4=625$ possible association patterns for $(x1\ x2\ x3\ x4)$. Let's assume three statistically significant association patterns are found: $(x1:3\ x2:2\ x3:5\ x4:4)$ $(x1:4\ x2:3\ x3:2\ x4:2)$ $(x1:5\ x2:4\ x3:3\ x4:2)$. Let's further assume the following probability information related to the three significant patterns just shown is available as below:

$$\begin{aligned} Pr(x1:3\ x2:2\ x3:5\ x4:4) &= 0.03 & Pr(x1:4\ x2:3\ x3:2\ x4:2) &= 0.05 \\ Pr(x1:5\ x2:4\ x3:3\ x4:2) &= 0.07 & Pr(x1:3) &= 0.15 & Pr(x1:4) &= 0.37 & Pr(x1:5) &= 0.23 \\ Pr(x2:2) &= 0.13 & Pr(x2:3) &= 0.35 & Pr(x2:4) &= 0.14 & Pr(x3:2) &= 0.27 \\ Pr(x3:3) &= 0.17 & Pr(x3:5) &= 0.3 & Pr(x4:2) &= 0.45 & Pr(x4:4) &= 0.12 \end{aligned}$$

Note that the degree of freedom of a joint probability model $Pr(x1\ x2\ x3\ x4)$ is $5^4 (=625) - 1 - 14$ (# of constraints) = 610. Therefore, there are multiple probability models that can satisfy the conditions. The process of model discovery is beyond the scope of this paper. Readers interested in further details are referred to chapter 9 of our book [10]. Nonetheless, we show one such probability model that is locally optimized to minimize the bias to unknown information:

$$\begin{aligned} Pr(x1:1\ x2:1\ x3:2\ x4:2) &= 0.057272747 & Pr(x1:1\ x2:2\ x3:5\ x4:1) &= 0.057272717 \\ Pr(x1:1\ x2:3\ x3:1\ x4:1) &= 0.12454547 & Pr(x1:1\ x2:4\ x3:5\ x4:4) &= 0.010909086 \\ Pr(x1:3\ x2:1\ x3:2\ x4:1) &= 0.12 & Pr(x1:3\ x2:2\ x3:5\ x4:4) &= 0.03 \\ Pr(x1:4\ x2:1\ x3:1\ x4:2) &= 0.09727271 & Pr(x1:4\ x2:1\ x3:5\ x4:1) &= 0.026363678 \\ Pr(x1:4\ x2:3\ x3:2\ x4:2) &= 0.05 & Pr(x1:4\ x2:3\ x3:5\ x4:2) &= 0.1754545 \\ Pr(x1:4\ x2:4\ x3:3\ x4:1) &= 0.020909093 & Pr(x1:5\ x2:1\ x3:3\ x4:4) &= 0.079090910 \\ Pr(x1:5\ x2:2\ x3:2\ x4:1) &= 0.042727260 & Pr(x1:5\ x2:4\ x3:1\ x4:1) &= 0.038181823 \\ Pr(x1:5\ x2:4\ x3:3\ x4:2) &= 0.07 \end{aligned}$$

Where the remaining $Pr(x1\ x2\ x3\ x4)$ s equal to 0.

The significance of an optimal probability model just shown is that it preserves the statistical properties of the significant association patterns while minimizing bias. In other words, the probability information of the model will reveal the statistically significant association patterns that define an access signature. This optimal probability model will be referred to as a reference model for a user.

5 Chi-square Goodness of Fit for Intrusion Detection

To determine masquerader intrusion, the command execution history will be examined in a regular time interval. If the command execution history of a user within some time interval could not produce a matching access signature with sufficient confidence level, then it will serve as a basis to suspect the existence of a masquerader intrusion. In the statistical inference framework, Chi-square test statistic λ^2 is used to determine the goodness of fit between the access signature revealed in the command execution history and that in the reference model. Specifically, we test the following null hypothesis versus the alternative hypothesis:

Null Hypothesis:

Masquerader intrusion exists if $\lambda^2 = \sum_{i=1}^k (o_i - e_i)^2 / e_i > \chi^2_{(1-\alpha, d)}$ where
 k is the number of significant patterns $\{ssp_i: i=1 \dots k\}$ revealed in the source model,
 N is the size of the command execution history within some given time interval,
 $e_i = N \cdot Pr_{source}(ssp_i)$ is the expected count of the i^{th} pattern ssp_i derived from source model,
 $o_i = N \cdot Pr_{target}(ssp_i)$ is the observed count of the i^{th} pattern ssp_i derived from target model,
 $\chi^2_{(1-\alpha, d)}$ is the value of the Chi-square random variable with a degree of freedom $d = k - 1$; where $0 < \alpha < 1$ is the significance level.

Alternative Hypothesis:

Masquerader intrusion does not exist if $\lambda^2 = \sum_{i=1}^k (o_i - e_i)^2 / e_i \leq \chi^2_{(1-\alpha, d)}$

In the formulation just shown, if ssp_i s are statistically significant patterns revealed in the reference model, then the source model is the reference model described in the previous section. The target model is then the probability distribution estimated by observing the actual frequency count of the occurrence of the ssp_i s in the command execution history within the given time interval. On the other hand, if ssp_i s are significant patterns revealed in the data block pertaining to the command execution history, then the source model is the probability distribution estimated from the data block while the target model is the reference model described in the previous section.

In other words, the goodness of fit statistical test will be conducted twice. First, it will be from the perspective of the reference model by comparing the access signature of the data block against its own signature. Second it will be from the perspective of the access data block by comparing the access signature in the reference model against its own signature. By combining the result of the statistical inference tests, a strong conclusion on the existence of masquerader intrusion could be drawn if the null hypothesis survives in both tests, or a weak conclusion on the existence of masquerader intrusion could be drawn if the null hypothesis survives in at least one test.

6 Experimental Study and Result Discussion

The proposed signature-based approach for masquerader intrusion is evaluated using the masquerade data available in the public domain for comparing various intrusion detection methods [11]. The masquerade data consists of 50 groups — each group is the access data of one user. Each data group for a user contains 15000 commands. The first 5000 commands within each data group do not contain any masqueraders and are used for the training purposes. The remaining 10000 commands are divided into 100 blocks. Within each block there are 100 commands that are used for testing purposes. The data blocks for testing are seeded with user masquerade that simulates masquerader intrusion. The location of masquerader intrusion is also provided in the masquerade data for evaluation purposes. About 5% (231 out of 5000) of the test data contain masquerades.

The masquerade data also provide additional information. At any given block after the initial 5000 commands a masquerade starts with a probability of 1%. If the previous block was a masquerade, the next block will also be a masquerade with a

probability of 80%. However, this information is not used in this experimental study. The following procedure is applied to each one of the 50 users in the study:

Step 1: Reproduce the data column of the first 5000 commands by shifting the data by one time unit. Repeat it 4 times to get a 4-column data table in which each row is a sequential pattern of 4 Unix commands appearing in the command execution history.

Step 2: Extract the first 5000 rows of the four-column data. Apply the function made available elsewhere [10] that implements the two criteria described in section 3 to discover statistically significant patterns. This set of statistically significant patterns defines the access signature of a user.

Step 3: Based on the set of the statistically significant patterns that defines the access signature, define the constraint set similar to that described in section 4.

Step 4: Apply the model discovery utility made available elsewhere [10] to derive the probability reference model for a user. This probability reference model, Pr_{ref} , reveals the access signature and has the (local) minimum biased information.

Step 5: For each one of the 100 test data block that has been enumerated into four-column data, apply the following procedure:

Step 5a: Reproduce the data column of the 100 Unix commands in the block by shifting the data by one time unit. Repeat it four times to get a four-column data table similar to that described in step 1.

Step 5b: Apply the function made available elsewhere [10] that implements the two criteria described in section 3 to discover statistically significant patterns. This set of statistically significant patterns defines the access signature of the test data.

Step 5c: For each statistically significant pattern c_i of a target user, derive the observed count o_i . Likewise, derive the expected count e_i as described in section 5.

Step 5d: Apply statistical inference based on Chi-square goodness of fit as described in section 4 to determine whether masquerader intrusion exists.

Step 6: Derive the correct detection rate, the false positive rate, and the false negative rate based on the result of the 100 test data blocks in step 5.

In order to determine the effectiveness of the approach, Receiver Operating Characteristic (ROC) curve [12] analysis is used to evaluate the result. The followings are the parameters used in a ROC curve analysis:

AP = Actual total positive counts in the test data (masquerader intrusion)

AN = Actual total negative counts in the test data (no masquerader intrusion)

PP = Number of predicted true positive counts

PF = Number of predicted false positive counts

FP = False positive rate = PF/AN

TP = True positive rate = PP/AP

An ROC curve is a graphical plot of FP (X-axis) against TP (Y-axis). Note that both FP and TP are between 0 and 1. An ideal intrusion detector will have a performance where TP = 1 and FP = 0; i.e., every masquerader intrusion is accurately captured with no false alarm. When TP = 1 and FP = 0, it also implies that there is no

false negative (since $TP = 1$) and all negative counts in the test data are correctly concluded by the detection system as no intrusion.

Referring to the threshold value $\chi^2_{(1-\alpha,d)}$ defined in section 4, FP and TP will vary with different choices of α . An ROC curve shows the changes of TP vs. FP as different threshold values are applied. An intrusion/anomaly detector is optimized if its threshold value $\chi^2_{(1-\alpha,d)}$ yields a point (FP, TP) that has the shortest distance to (0,1) in an ROC curve.

An ROC curve is derived for every single user based on the six steps described previously. An ROC curve using all the data — referred to as overall ROC — is also derived to illustrate the overall performance. In the case of overall ROC, all 5000 blocks (100 blocks for each of the 50 users) are used as testing data. Again, an overall ROC curve is obtained by varying the threshold $\chi^2_{(1-\alpha,d)}$.

Referring to section 4, a Chi-square test statistic λI^2 could be derived by using the training data as the source, and the testing data as the target. Likewise, another Chi-square test statistic $\lambda 2^2$ could be derived by using the testing data as the source, and the training data as the target. We then derive an overall Chi-square test statistic λ^2 based on the linear combination of λI^2 and $\lambda 2^2$; i.e., $\lambda^2 = (1-w) \cdot \lambda I^2 + w \cdot \lambda 2^2$. The choice of w varies from 0 to 1 with an increment of 0.1. In applying the statistical inference described in section 4 using the test statistic $\lambda^2 = (1-w) \cdot \lambda I^2 + w \cdot \lambda 2^2$, the optimal setting for w is 0.1. Using the test statistic λ^2 , the overall ROC curve and the ROC curves for the 50 users (but skipping those with a testing data set that has no intrusion) are shown in Fig.1.

Fig. 2 shows the ROC band envelope that encloses all the ROC curves, the overall ROC curve, and the *estimated* ROC curve. The *estimated* ROC curve is based on “averaging” all ROC curves. Fig. 2 also shows the ROC curves that are one and two standard deviation away from the estimated ROC curve. In Fig. 2, one could note that the ROC band is wide due to a wide variation across all 50 users. Consequently, it is no surprise that the estimated ROC matches closely to the overall ROC curve only partially at $FP < 0.2$ or $FP > 0.8$.

Fig. 3 and Fig. 4 show the ROC curves of different selected users. Fig. 5 shows the ROC curve for six different approaches reported elsewhere [7]. Fig. 5 is reproduced for gaining insights into achievable performance. An interesting observation in comparing Fig. 1 and Fig. 3 is that the overall optimal performance for 50 users is better than that for 8 selected users as shown in the corresponding ROC curve. But by comparing Fig. 3 and Fig. 4, the optimal performance for 6 selected users is better than that of all 50 users and 8 selected users. In other words, one must be mindful that performance comparison is only meaningful when the ROC curves generated for different methods are based on the same population of sample users.

One final note about the experimental result is that only normal event/behavior instances are available in the training data set for deriving access signature and reference model. If we are willing to reduce the size of the testing data, it is conceivable to include some of the masquerade intrusion test data as training data to explore the idea of incorporating both access signature and intrusion signature in a reference model. To extend the signature-based approach to incorporate intrusion signature, we only need to modify the statistical hypothesis test by introducing two

additional test statistics in addition to λI^2 and $\lambda 2^2$ described earlier to account for the consideration of known intrusion patterns. This additional study will be included in our next report.

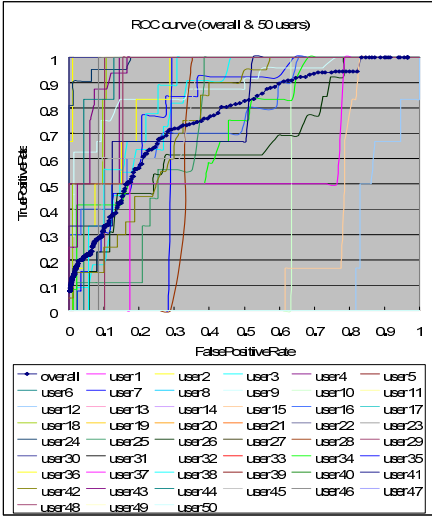


Fig. 1. ROC curves of all 50 users

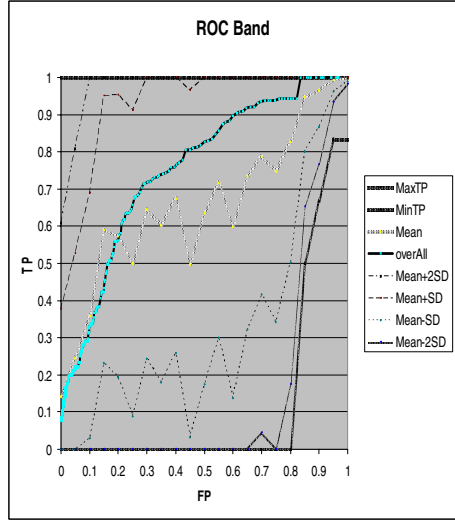


Fig. 2. ROC band and estimated ROC curve

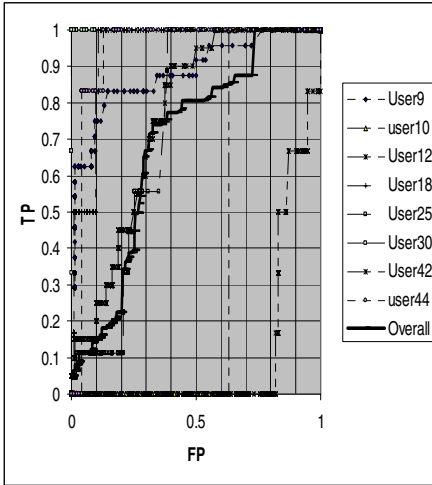


Fig. 3. ROC curve of 8 selected users

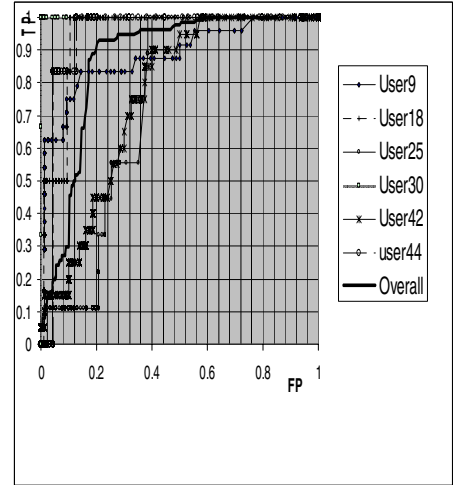


Fig. 4. ROC curve of 6 selected users

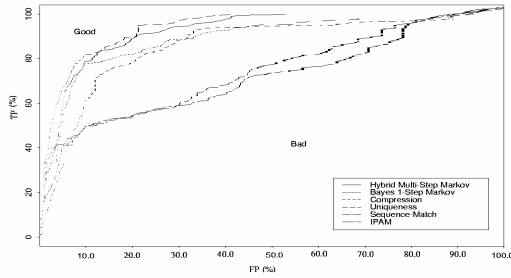


Fig. 5. ROC curves for six different approaches

7 Conclusion

A signature-based approach is presented for discovering masquerader intrusion. In this proposed approach we introduce the concept of an access signature, which is a collection of statistically significant association patterns. The concept of an access signature is appealing because it allows one to derive a probability model that captures the uniqueness of the access behavior of a user while taking into the consideration of the intra-usage variation. Equally important, the derived probability model provides a basis for detecting masquerader intrusion efficiently. As shown in this paper, efficient detection on masquerader intrusion is simply a process of matching the real time online access signature against the one in the probability reference model based on Chi-square statistical test for goodness of fit. The experimental study also shows an encouraging result in the comparative evaluation. Although we focus on this paper only the masquerader intrusion, the signature-based approach is extensible for incorporating intrusion signatures, as well as for discovering other kinds of intrusion; e.g., network intrusion. This will be the focus of our future research.

Acknowledgement. The author is indebted to the anonymous reviewers who offered many valuable comments that resulted in the improvement of this manuscript. This work is supported in part by a NSF DUE CCLI grant #0088778, and a PSC-CUNY Research Award. Students in the author's Data Mining class of Fall/2004 semester have contributed significantly to the experimental study.

References

1. Sandeep Kumar, *Classification and Detection of Computer Intrusions*, Ph.D. thesis, Purdue University, August 1995.
2. Wenke Lee and Salvaor Srolfo, "Data Mining Approaches for Intrusion Detection," *Proc. of the 7th USENIX Security Symposium*, San Antonio, Texas, Jan., 1998.
3. *etrust Audit: Policy Management Guide 1.5*, Computer Associates, 2003.
4. Sun Microsystems. *SunShield Basic Security Module Guide*.

5. Jeremy Frank, "Artificial Intelligence and Intrusion Detection: Current and Future Directions," June 9, 1994
6. W.-H. Ju and Y. Vardi, "A Hybrid High-order Markov Chain Model for Computer Intrusion Detection," *J. of Computational & Graphical Statistics*, V. 10(2), 2001.
7. M. Schonlau, W. Dumouchel, W.-H., Ju, A.F. Karr, M. Theus, Y. Vardi, "Computer Intrusion: Detecting Masquerades," *Statistical Science*, V. 16, #1, 58-74, 2001.
8. R. Agrawal, T. Imielinski, A. Swami, "Mining Association Rules between Sets of Items in large Databases," *Proc. ACM SIGMOD Conf.*, Washington DC, May 1993.
9. Bon Sy, "Discovering Association Patterns based on Mutual Information," Machine Learning and Data Mining in Pattern Recognition (editor: Petra Pernert), Lecture Notes in Artificial Intelligence, Springer-Verlag, July 2003.
10. Bon Sy and Arjun Gupta, *Information-statistical Data Mining: Warehouse Integration with Examples of Oracle Basics*, ISBN 1-4020-7650-9, 2004.
11. <http://www.schonlau.net/intrusion.html>
12. T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Data Mining Researchers," *Technical Report HPL-2003-4*, Intelligent Enterprise Technologies Laboratory, HP Laboratories Palo Alto, Jan 7, 2003.
13. E. Eskin, "Anomaly Detection over Noisy Data Using Learned Probability Distributions," Proc. of the 17th International Conference on Machine Learning, 2000, pp 255-262, Morgan Kaufmann, San Francisco, CA.
14. F. Esponda, S. Forrest, and P. Helman, "A Formal Framework for Positive and Negative Detection," *IEEE Transactions on Systems, Man and Cybernetics*. 34:1 pp. 357-373 (2004).

Discovery of Hidden Correlations in a Local Transaction Database Based on Differences of Correlations

Tsuyoshi Taniguchi, Makoto Haraguchi, and Yoshiaki Okubo

Division of Computer Science, Hokkaido University,
N-14 W-9, Sapporo 060-0814, Japan
{tsuyoshi, makoto, yoshiaki}@kb.ist.hokudai.ac.jp

Abstract. Given a transaction database as a global set of transactions and its sub-database regarded as a local one, we consider a pair of itemsets whose degrees of correlations are higher in the local database than in the global one. If they show high correlation in the local database, they are detectable by some search methods of previous studies. On the other hand, there exist another kind of paired itemsets such that they are not regarded as characteristic and cannot be found by the methods of previous studies but that their degrees of correlations become drastically higher by the conditioning to the local database. We pay much attention to the latter kind of paired itemsets, as such pairs of itemsets can be an implicit and hidden evidence showing that something particular to the local database occurs even though they are not yet realized as characteristic ones. From this viewpoint, we measure paired itemsets by a difference of two correlations before and after the conditioning to the local database, and define a notion of DC pairs whose degrees of differences of correlations are high. As the measure is non-monotonic, we present an algorithm, searching for DC pairs, with some new pruning rules for cutting off hopeless itemsets. We show by an experimental result that potentially significant DC pairs can be actually found for a given database and the algorithm successfully detects such DC pairs.

1 Introduction

In the studies of data mining from transaction databases, many studies have been paying much attention to finding itemsets with high supports, paired itemsets appeared in association rules with high confidence [1], or paired itemsets with strong correlation [6, 7, 8, 9]. These notions are considered useful for distinguishing characteristic itemsets from other ones in a single transaction database. A similar strategy based on the notion of change of supports, known as Emerging Patterns [2, 3], is successful even for finding itemsets characterizing either of two databases. All of the notions about itemsets are thus proposed to extract (paired) itemsets required to be characteristic in a given database or either of a given pair of databases.

However, as has been indicated in the study of Chance Discovery [10], some itemsets not characteristic in the above sense are also useful, as they are *potentially significant* under some condition. For instance, suppose we have a transaction database for supermarkets in a particular area and the database includes the information of ages of customers and goods on sale as items. We here consider the problem of capturing some correlations between some ages and some goods in the database. We regard the correlations as an interest of customers of some ages in some goods. For example, consider a case that degrees of the correlations are not high in a particular area but are very low in a global area including the particular area. The correlation cannot be found by search methods of previous studies because the degrees of the correlations are not high in both global and particular areas. However, there is a possibility that the customers of the ages are interested in the goods in the particular area more than in the global area by some factor even if the correlations in the particular area are not regarded as characteristic. It may be worth remarking this specific phenomenon as an implicit and hidden evidence in order to consider a new strategy for sale. Moreover, consider a case that the database includes the information of time as item. In the particular area, we can find characteristic correlations with high degree of correlation in time t_1 or t_2 after t_1 by search methods of previous studies. But we may want to know an implicit correlation which may become a characteristic correlation in t_3 after t_2 . In short, we want to know customers of some ages start to be interested in some goods. In the case, non-characteristic correlations in t_2 with high degrees of differences of correlations from t_1 to t_2 may be useful.

From the viewpoints mentioned in the above, for a given global database and its local database obtained by a certain conditioning, the purpose of this paper is to present an algorithm for finding pairs of itemsets such that (1) the paired itemsets are not necessarily characteristic, where we say that two itemsets are characteristic in a database if the correlation between them is high, (2) the degrees of correlation become much higher in the local database than ones in the global database. That is, we are going to observe the degrees of difference of correlations before and after the conditioning to the local database. Such a pair of itemsets with high degrees of difference of correlations is called a DC pair. We confirm by an experiment that potentially significant DC pairs can be actually found for a given database.

It is generally a hard problem to find DC pairs, as the degrees of difference of correlations are never monotonic w.r.t. the standard ordering of itemsets, namely the set inclusion. For this reason, we consider a restricted problem under given two parameters, ζ and ϵ . More precisely speaking, we evaluate the degrees of difference of correlations by a function defined with ζ and ϵ and restrict DC pairs we try to find. Then, we prove that a monotone property over itemsets can be observed in the mining of DC pairs depending on ζ or ϵ . Based on this monotone property, we can design some pruning rules for cutting off hopeless itemsets X and Y not satisfying the constraints of DC pairs.

1.1 Related Works and Paper Organization

There exist many works in the field of data mining that are based on a strategy of contrasting two or more databases in order to extract significant properties or patterns from a huge data set. Particularly, data mining techniques, known as contrast-set mining [2, 3, 4, 5], have been designed specifically to identify differences between databases to be contrasted.

For instance, in the study of Emerging Patterns [2, 3] for two transaction databases, itemsets whose supports are significantly higher in one database than in another one are considered significant, as they can be candidate patterns for distinguishing the former from the latter. A similar strategy is also used in the system STUCCO [4] in order to obtain characteristic itemsets in one database based on χ^2 test. In addition, the system, Magnum Opus [5], examines relations between itemsets and a database among several databases. On the other hand, what this paper tries to find are paired itemsets whose correlations drastically increase in one database. Thus we can say that the subject of this paper is a kind of "contrast-set mining of correlations between itemsets".

Secondly, many methodologies have been proposed to detect characteristic correlations in a single database [6, 7, 8]. In these studies, using some function measuring the degree of correlation between itemsets, strongly correlated itemsets in a given database or in one database from given two databases are examined. Thus, these methods are also used to discover itemsets or family of itemsets that are characteristic in one database. On the other hand, the algorithm presented in this paper is designed so as to find even paired itemsets whose correlation in one database is not significantly high but is significantly higher than correlation in another database. Our algorithm may find the characteristic paired itemsets as special cases, but is never supposed to find only characteristic ones. To find these paired itemsets, we present in this paper some new pruning rules so that the algorithm successfully detects even non-characteristic paired itemsets.

Finally, several notions about correlations have been proposed and used in the above previous studies from information theoretic or statistical viewpoints, then we describe our standpoint that we use a measure to evaluate correlations. If we need to consider even negative events that itemsets do not appear in transactions, the notion of correlations based on χ^2 -test shall be taken into account. But this paper is based on the notion of self mutual information without taking log to measure positive relationships between events that itemsets occur.

The rest of this paper is organized as follows. The next section defines some terminologies used throughout this paper. In Section 3, we introduce the notion of DC pairs and define our problem of mining DC pairs. An algorithm for finding DC pairs is described in Section 4. Section 5 presents our experimental results. In the final section, we summarize our study and discuss future work.

2 Preliminaries

Let $\mathcal{I} = \{i_1, i_2, \dots, i_n\}$ be a set of *items*. An *itemset* is a subset of \mathcal{I} . A *transaction database* \mathcal{D} is a set of transactions, where a transaction is an itemset.

We say that a transaction t *contains* an itemset X , if $X \subseteq t$. For a transaction database \mathcal{D} and an itemset X , the *occurrence* of X over \mathcal{D} , denoted by $O(X, \mathcal{D})$, is defined as $O(X, \mathcal{D}) = \{t | t \in \mathcal{D} \wedge X \subseteq t\}$, and the *probability* of X over \mathcal{D} , denoted by $P(X)$, is defined as $P(X) = |O(X, \mathcal{D})|/|\mathcal{D}|$.

For an itemset C , a *sub-database* of \mathcal{D} w.r.t. C , denoted by \mathcal{D}_C , is defined as the set of transactions containing C in \mathcal{D} , that is, $\mathcal{D}_C = O(C, \mathcal{D})$. The *complement* of \mathcal{D}_C w.r.t. \mathcal{D} is denoted by $\overline{\mathcal{D}_C}$ and is defined as $\overline{\mathcal{D}_C} = \mathcal{D} - \mathcal{D}_C$.

For itemsets X and Y , the *correlation* between X and Y over a transaction database \mathcal{D} , $correl(X, Y)$, is defined as $correl(X, Y) = P(X \cup Y)/P(X)P(Y)$. For a sub-database \mathcal{D}_C , the correlation between X and Y over \mathcal{D}_C , $correl_C(X, Y)$, is given by $correl_C(X, Y) = P(X \cup Y|C)/P(X|C)P(Y|C)$, where $P(X|C) = P(X \cup C)/P(C)$. Note here that correlations are defined for only itemsets X whose supports in \mathcal{D} and \mathcal{D}_C are non-zero. We regard a pair of X and Y such that $correl(X, Y) > 1$ as characteristic since $P(X|Y) > P(X)$ holds. Notice that $P(Y|X) > P(Y)$ holds, too. Similarly, we regard a pair of X and Y such that $correl(X, Y) \leq 1$ as non-characteristic.

3 DC Pair Mining Problem

In this section, we define a notion of DC pairs and our problem of mining them.

For a pair of itemsets X and Y , we especially focus on “difference of correlations observed by conditioning to the local database”. The difference of correlations is measured by the following ratio:

$$change(X, Y; C) = \frac{correl_C(X, Y)}{correl(X, Y)} = \frac{P(C)P(C|X \cup Y)}{P(C|X)P(C|Y)}. \quad (1)$$

Let $\rho (> 1)$ be an admissible degree of difference of correlations. In our framework, a pair of itemsets X and Y is considered significant if $change(X, Y; C) \geq \rho$ holds. Since we assume C is given by users, $P(C)$ can be regarded as a constant. Therefore, the change is actually evaluated with the following function g :

$$g(X, Y; C) = \frac{P(C|X \cup Y)}{P(C|X)P(C|Y)}. \quad (2)$$

A pair of itemsets X and Y is called a *DC pair* if $g(X, Y; C) \geq \rho/P(C)$. We try to find all DC pairs efficiently. It should be noted here that the function g behaves *non-monotonically* according to expansion of itemsets X and Y . So we cannot apply a simple pruning method like one Apriori adopted [1]. Therefore, we approximate the above problem according to the following naive strategy:

Find pairs of X and Y which give higher values of $P(C|X \cup Y)$, keeping the values of $P(C|X)$ and $P(C|Y)$ small.

With a new parameter ζ ($0 \leq \zeta \leq 1$), our approximated problem is precisely defined as follows:

Definition 1. DC Pairs Mining Problem

Let C be an itemset for conditioning. Given ρ and ζ , DC pair mining problem is to find any pairs of X and Y such that $P(C|X \cup Y) > \zeta$, $P(C|X) < \epsilon$ and $P(C|Y) < \epsilon$, where $\epsilon = \sqrt{\zeta \cdot P(C)/\rho}$.

4 Algorithm for Finding DC Pairs

In this section, we present an algorithm to solve the DC pair mining problem. In Section 3, by using parameters ζ and ϵ , we restrict DC pairs we try to find. But, $P(C|Z)$ behaves *non-monotonically* according to expansion of an itemset Z as well as g . This means that there is a possibility that we have to examine all itemsets in database since a simple pruning method cannot be used. Then, we prove some pruning rules by considering the problem of mining DC pairs in top-down manner. Therefore, in this paper, we explain an algorithm which candidates Z for compound itemsets of DC pairs such that $P(C|Z) > \zeta$ are found at first in top-down manner. In order to examine itemsets in top-down manner, we firstly enumerate maximal itemsets in the local database \mathcal{D}_C because $P(Z|C) > 0$ must hold. After all, the computation for mining DC pairs is divided into two phases:

Phase1: Identifying Candidates for Compound Itemsets

An itemset Z such that $P(C|Z) > \zeta$ is identified as a candidate itemset from which DC pairs X and Y are obtained as $Z = X \cup Y$.

Phase2: Dividing Compound Itemsets

Each candidate Z is divided into two itemsets X and Y such that $Z = X \cup Y$, $P(C|X) < \epsilon$ and $P(C|Y) < \epsilon$.

In the algorithm, there is a case that some candidate Z may not be decomposable. Therefore, we consider checking the possibility for Z to be divided into some DC pair. Then, we first describe a basic enumeration schema, and then introduce more refined one taking the decomposability into account.

4.1 Pruning Search Branches by Dropping Items

For each maximal itemset Z_{max} found in \mathcal{D}_C , we first examine Z_{max} , then its proper subsets are examined, and so on. During this search, we can prune useless branches (itemsets) based on the following observation.

Let Z be an itemset containing an item i . Suppose that there exists a subset Z' of Z such that $i \in Z' \subset Z$ and $P(C|Z') > \zeta$. Since $P(C|Z') = P(C)P(Z'|C)/P(Z') > \zeta$, $P(Z'|C) > \zeta \cdot P(Z')/P(C)$ holds. Therefore, $P(i|C) \geq P(Z'|C) > \zeta \cdot P(Z')/P(C) \geq \zeta \cdot P(Z)/P(C)$. As the result, we have $P(C \cup i) > \zeta \cdot P(Z)$. This means that if $P(C \cup i) \leq \zeta \cdot P(Z)$ holds, then we cannot obtain any subset Z' of Z containing i such that $P(C|Z') > \zeta$. That is, assuming Z as a search node in Phase1, if $P(C \cup i) \leq \zeta \cdot P(Z)$ holds, any immediate subset of Z containing i does not have to be examined. Therefore, we can safely drop i from Z .

Dropping Items:

For a search node (itemset) Z and an item $i \in Z$, if $P(C \cup i) \leq \zeta \cdot P(Z)$, any subset Z' containing i never be a child node of Z in our top-down construction process. In other words, any child node consists of only items in Z that are not dropped.

As a special case, if any item $i \in Z$ is dropped, we do not need to examine any subset of Z .

Termination Condition:

For a search node (itemset) Z , if $\max\{P(C \cup i) | i \in Z\} / \zeta \leq P(Z)$ holds, then Z does not have to be expanded further.

The termination condition provides a theoretical lower bound of our search in Phase1. Since $i \in Z$ and $P(Z|C) > 0$, then $P(i|C) > 0$ holds. Therefore, we can obtain the following.

Lower Bound of Search in Phase1:

If a search node Z is visited in Phase1, then $P(Z) \leq \max p_\zeta$, where $\max p_\zeta = \max\{P(C \cup i) | P(i|C) > 0\}$. In other words, any search node Z whose probability exceeds $\max p_\zeta$ never be generated in Phase1.

4.2 Pruning Search Branches Based on Decomposability

The pruning mechanism just discussed above can become more powerful by taking some constraint in Phase2 into account. More concretely speaking, we can perform the operation of “Dropping Items” more frequently.

In Phase2, each candidate Z found in Phase1 is divided into two itemsets X and Y such that $P(C|X) < \epsilon$ and $P(C|Y) < \epsilon$. Similar to the above discussion, for any $i \in X \cup Y (= Z)$, $P(C \cup Z) < \epsilon \cdot P(i)$ holds. Therefore, if there exists an item $i \in Z$ such that $P(C \cup Z) \geq \epsilon \cdot P(i)$, Z cannot be divided into two parts satisfying the constraint on ϵ . In other words, such an item i never be a member of adequate two parts. Therefore, i can be dropped from Z . Thus, we can obtain a revised operation on search nodes which is more powerful.

Dropping Items (Revised):

For a search node Z and an item $i \in Z$, if $P(C \cup i) \leq \zeta \cdot P(Z)$ or $P(i) \leq P(C \cup Z) / \epsilon$, i can be dropped from Z .

According to it, a new termination condition and a new theoretical lower bound is given as follows:

Termination Condition (Revised):

For a search node Z , if $\max\{P(C \cup i) | i \in Z\} / \zeta \leq P(Z)$ or $\max\{P(i) | i \in Z\} \leq P(C \cup Z) / \epsilon$, then Z does not have to be expanded further.

Lower Bound of Search in Phase1 (Revised):

If a search node Z is visited in Phase1, then $P(Z) \leq \max p_\zeta$ and $P(C \cup Z) \leq \epsilon \cdot \max p_\epsilon$ holds, where $\max p_\epsilon = \max\{P(i) | P(i|C) > 0\}$.

4.3 Another Termination Condition in Phase1

In this section, we show another lower bound in Phase1 by taking the complement $\overline{\mathcal{D}_C} = \mathcal{D} - \mathcal{D}_C$ into account. We expect this lower bound stop expanding search nodes before Dropping Items start to work.

Suppose a DC pair of X and Y is obtained from an itemset Z , that is, $Z = X \cup Y$. Then, $P(C|Z) = |O(Z, \mathcal{D}_C)| / (|O(Z, \mathcal{D}_C)| + |O(Z, \overline{\mathcal{D}_C})|) > \zeta$ and $P(C|X) = |O(X, \mathcal{D}_C)| / (|O(X, \mathcal{D}_C)| + |O(X, \overline{\mathcal{D}_C})|) < \epsilon$. Then it follows that $|O(Z, \mathcal{D}_C)| > \frac{\zeta}{1-\zeta} |O(Z, \overline{\mathcal{D}_C})|$ and $|O(X, \mathcal{D}_C)| < \frac{\epsilon}{1-\epsilon} |O(X, \overline{\mathcal{D}_C})|$. Therefore, $\frac{\zeta}{1-\zeta} |O(Z, \overline{\mathcal{D}_C})| < |O(Z, \mathcal{D}_C)| \leq |O(X, \mathcal{D}_C)| < \frac{\epsilon}{1-\epsilon} |O(X, \overline{\mathcal{D}_C})|$. As a result, we have $|O(Z, \overline{\mathcal{D}_C})| < k(\zeta, \epsilon) |O(X, \overline{\mathcal{D}_C})|$, where $k(\zeta, \epsilon) = \frac{(1-\zeta)\epsilon}{\zeta(1-\epsilon)}$. Furthermore, as $|O(Z, \overline{\mathcal{D}_C})| \leq |\overline{\mathcal{D}_C}| \leq |\mathcal{D}|$, we have $|O(Z, \overline{\mathcal{D}_C})| < k(\zeta, \epsilon) |\mathcal{D}|$. Conversely, if $|O(Z, \overline{\mathcal{D}_C})| \geq k(\zeta, \epsilon) |\mathcal{D}|$, it follows that Z as well as any subset Z' of Z is never decomposable to obtain DC pairs, as $|O(Z', \overline{\mathcal{D}_C})| \geq |O(Z, \overline{\mathcal{D}_C})|$.

Termination Condition Based on Complement:

If $|O(Z, \overline{\mathcal{D}_C})| / |\mathcal{D}| \geq k(\zeta, \epsilon)$, Z does not need to be expanded further.

In the top-down mining process of DC pairs, we firstly check the above termination condition for the present itemset Z . If the condition does not hold, then we make the next node Z' with the help of the rule of dropping items.

4.4 Dividing Compound Itemsets

In Phase 2, we divide a candidate compound itemset Z into itemsets X and Y such that $Z = X \cup Y$, $X \cap Y = \emptyset$, $P(C|X) < \epsilon$, and $P(C|Y) < \epsilon$. For this purpose, we consider a lattice of itemsets with Z as its greatest itemset, and enumerate $X \subset Z$ in a bottom-up manner, from a singleton itemset to Z , with the following pruning rule.

Dropping Items in Phase 2:

For a search node (itemset) X and an item $i \in X$, if $P(C \cup Z) \leq \epsilon P(i \cup X)$, any superset of X containing i does not need to be expanded further.

The above rule is exactly dual to the rule of Dropping Items in Phase1, and is therefore similarly proved and utilized for cutting off useless branches to next nodes including items that can be dropped.

5 An Experiment

In this section, we present some experimental results on the mining of DC pairs. The main purpose of experiments is to confirm that potentially significant DC pairs can be actually found for a given database.

5.1 Datasets and Implementation

At first, we explain a database we use in our experiment. We carried out the experiments on Entree Chicago Recommendation Data, a family of databases from

the UCI KDD Archive (<http://kdd.ics.uci.edu>). It consists of eight databases each of which contains restaurant features in a region, e.g. Atlanta, Los Angeles, New Orleans and so on in the USA. To examine DC pairs given a particular region to be compared with the whole regions, we consider a new item working as a name for each region, and assign it to every transaction of the corresponding database. By this operation, we have an integrated database of 4160 transactions and 265 items. The items represent various restaurant features as "Italian", "romantic", "parking" and so on. Given the integrated database, we have developed a system written in C for finding DC pairs. All experiments are conducted on 1.5 GHz PentiumIV PC with 896 MB memory.

As we have already explained in Section 4, our top-down search procedure enumerates compound itemsets Z such that $P(C|Z) > \zeta$, starting from maximal itemsets in \mathcal{D}_C and using the pruning rules based on Dropping Items (Revised) and two Termination Conditions.

We carried out a preliminary experiment before the experiment at first. So, we can know our pruning rules are difficult to work well when the size of an itemset examined is long. We describe the reason in 5.2. Therefore, let the purpose of the experiment be to confirm that potentially significant DC pairs can be actually found for a given database and the algorithm successfully detects such DC pairs and to examine a performance of our pruning rules when the size of an itemset examined is short. Moreover, based on the result of the experiment, we examine a possibility of an efficient search of DC pairs.

For the above purpose, we here assume that our search procedure starts from itemsets of shorter length than maximal itemsets in \mathcal{D}_C . More precisely speaking, instead of maximal itemsets in \mathcal{D}_C , we introduce a family of itemsets such that (1) the lengths are no more than a given size parameter (6 in our experiment) and that (2) they are maximal among all itemsets having non-zero support and satisfying (1), where the order to define the maximality is also based on the set inclusion.

5.2 Experimental Results

Our experimental results are summarized in Figure 1, where ρ is the ratio of $correl_C(X, Y)$ to $correl(X, Y)$, ζ is a parameter in our search strategy, and $|N_{full}|$ is the number of itemsets in \mathcal{D}_C whose sizes are no more than the size parameter. ρ , ζ and size parameter are set for the values 3.0, 0.4 and 6, respectively in our experiment. $|N_{drop}|$ is the number of itemsets actually examined in Phase1, $|P(C|Z) > \zeta|$ denotes the number of itemsets Z such that $P(C|Z) > \zeta$ in \mathcal{D}_C whose sizes are no more than 6, $|DC|$ is the number of detected DC pairs. Finally, $|DC_{NotCor}|$ is the number of DC pairs of itemsets whose degree of correlation is less than or equal to 1.

There exist various kinds of DC pairs in the experimental data. For instance, in New Orleans, a DC pair $X = \{Entertainment, Quirky, Up\ and\ Coming\}$ and $Y = \{\$ 15-\$ 30, Private\ Parties, Spanish\}$ is found. The pair shows high degree of difference of correlations by conditioning to New Orleans. But since the pair shows very high degree of correlation as a result of its conditioning, the pair can

be found by search methods of previous studies. Also, in many cases, such DC pairs show high degrees of correlations in global database in the experiment. In short, such DC pairs may not be worth paying attention to by especially conditioning to New Orleans. On the other hand, there exists a pair $X = \{Quirky\}$ and $Y = \{Good\ Decor, Italian, \$15-\$30, Good\ Service\}$ in DC pairs in New Orleans. The pair is not correlated in both global database and local database. Therefore, the pair cannot be found by search methods of previous studies. But the pair shows high degree of difference of correlations by conditioning to New Orleans. In short, the pair shows not high degree of correlation in New Orleans, on the other hand, the pair shows very low degree of correlation in a global database. We pay much attention to such DC pairs. We consider such DC pairs can be useful in some cases. For instance, people who look for a restaurant in New Orleans may be interested in a "quirky Italian restaurant" which is a hidden feature in New Orleans in contrast with a "quirky Spanish restaurant" which is a significant feature in both global and local database because there may be some factor of its high degrees of difference of correlations even if the pair doesn't show high degree of correlation. As we described the above, it is shown that potentially significant DC pairs can be actually found for the given database and our algorithm detects such DC pairs. In addition, various potentially significant DC pairs are found in the experimental data.

As is shown in Figure 1, the number of compound itemsets examined is certainly reduced by the pruning rules in Section 4. Every pruning rule we have presented is theoretically safe in the sense that they cut off some branches only when it is proved that no solution can be reached through the branches. However, the degree of reduction does not seem sufficient to improve the efficiency. We consider the causes as follows.

The first cause is a low chance that our pruning rules can be applied to itemsets examined in our search. By a simple operation of our pruning rules, there is a possibility that we can turn out that many itemsets don't have to be examined. But our pruning rules cannot reduce so many itemsets examined in the experiment because there are not many opportunities that our pruning rules can be applied to the itemsets. So, we analyze a property of our pruning rules. And we can know our pruning rules are difficult to work well when a difference between a probability of an itemset Z and a probability of an item $i \in Z$ is large in a global or a local database. Note here that, in a sparse data which is often used in data mining, many itemsets whose size is long have a low probability and the difference is large in many cases. This is the cause that our pruning rules are difficult to be applied to the itemsets whose size is long. Therefore, in order to solve the problem and increase the chance of our pruning, we have to weaken conditions of our pruning rules and modify a procedure in our search.

The second cause is the large number of an itemset Z such that $P(C|Z) > \zeta$ in the experimental data. In Fig. 1, in Atlanta, it seems that our pruning rules cannot reduce only 100 thousands itemsets out of one million and 920 thousands all itemsets. But there are one million and 570 thousands itemsets Z such that $P(C|Z) > \zeta$ which we find in step 1. Therefore, there are only

$\rho = 3.0, \zeta = 0.4$							
region	$sup(C)$	ϵ	$ N_{full} $	$ N_{drop} $	$ P(C Z) > \zeta $	$ DC $	$ DC'_{NotCor} $
Atlanta	0.064	0.0922	1922264	1826678	1575575	112877	269
Los Angeles	0.107	0.118	1857501	1760522	1769705	30404	97
New Orleans	0.079	0.102	1120224	1071306	1027241	39158	55
San Francisco	0.100	0.114	2154595	2113443	1735822	134520	312

Fig. 1. Experimental Results

350 thousands itemsets which don't have to be examined in Step 1. Notice here that, in Los Angeles, the number of itemsets actually examined is less than the number of itemsets Z such that $P(C|Z) > \zeta$. This phenomenon is influenced by decomposability of DC pair described in 4.2. Then, by taking decomposability of DC pairs into account more, there is a possibility that itemsets examined can be reduced. We describe the prosperity of the above problems in Concluding Remarks.

6 Concluding Remarks

Given a transaction database \mathcal{D} and its sub-database \mathcal{D}_c , we proposed the notion of DC pairs. A pair of itemsets X and Y is called a DC pair if the correlation between X and Y in \mathcal{D}_c is relatively high to one in the original \mathcal{D} with some degree. It should be noted that the correlation is not always high in \mathcal{D}_c even though we can observe some degree of correlation change for \mathcal{D} and \mathcal{D}_c . In this sense, such a pair might not be characteristic in \mathcal{D}_c . Thus, DC pairs are regarded as *potential characteristics* in the sub-database. Our experimental results showed that DC pairs which are potentially significant can be actually found for ‘‘Entree Chicago Recommendation Data’’ under conditioning by each region. On the other hand, it is turned out that our pruning rules have to be more powerful before we apply our algorithm to a problem in a real life. Then, in order to search DC pairs efficiently, we have some prosperities as follows.

At first, we try to weaken conditions of our pruning rules and modify a procedure. In the experiment, we can know our pruning rules are difficult to work well when a difference between $P(Z)$ and $P(i)$ or $P(C \cup Z)$ and $P(C \cup i)$ ($i \in Z$) is large. Conversely, if the difference is small, our pruning rules can work well. So, in order to increase an opportunity of our pruning, a set of itemset whose probability is almost same can be useful. In order to make use of the set of itemsets, we have to weaken conditions of pruning rules. In short, for an itemset Z , our pruning rules need to be applied to itemsets $Z' \subset Z$ not items $i \in Z$. Moreover, in order to use the weaken rules, we have to modify a search procedure. Next, we try to take advantage of decomposability of DC pairs more. In 4.2, if an itemset Z examined doesn't contain $X(i \in X \subset Z)$ such that $P(C|X) < \epsilon$, we drop an item $i \in Z$ from Z because $Z'(i \in Z' \subset Z)$ cannot be divided into two itemsets X and Y such that $P(C|X) < \epsilon$ and $P(C|Y) < \epsilon$. Notice here that we can make use of the decomposability more.

Briefly speaking, a DC pair is a pair of itemsets X and Y . Therefore, if X cannot hold $P(C|X) < \epsilon$ or Y cannot hold $P(C|Y) < \epsilon$, a pair of X and Y is not a DC pair. In addition, when an itemset Z is divided into two itemsets X and Y ($Z = X \cup Y, X \cap Y = \emptyset$), X or Y contains an item $i \in Z$ necessarily. In short, Z cannot be divided into a DC pair if $X(i \in X \subset Z)$ cannot hold $P(C|X) < \epsilon$. In a preliminary experiment, by taking the new decomposability into account, there is a possibility that the number of itemsets examined may become no more than the half number without using the new decomposability. We are trying to tackle the above problems.

Finally, we discuss our future work. As we described in Introduction, we consider our frame work can be applied to time series data. In this paper, we pay attention to a difference of correlation observed by conditioning to the local database. Based on the notion of the DC pair, if we pay attention to a difference of correlation from time t_1 to t_2 after t_1 , our algorithm can be applied to time series data easily although we have to take the information particular to time series data into account. In this problem, characteristic correlation in t_1 or t_2 can be found by using search methods of previous studies. But there may be a case that we want to know an implicit correlation that may become characteristic in t_3 after t_2 although we have to consider an interval between t_1 and t_2 seriously. We can find such a correlation by capturing a difference of correlations from t_1 to t_2 . We are considering the application of the notion of the DC pair to time series data.

References

1. R. Agrawal, R. Srikant. Fast algorithms for mining association rules. In *Proc. of the Int'l Conf. on Very Large Data Bases*, pages 487-99, 1994.
2. G. Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. In *Proc. of the 5th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pages 43-52, 1999.
3. H. Alhammady and K. Ramamohanarao. Using emerging patterns and decision trees in rare-class classification. In *Proc. of the 4th IEEE Int'l Conf. on Data Mining*, pages 315-18, 2004.
4. S. D. Bay and M. J. Pazzani. Detecting group differences: mining contrast sets. *Data Mining and Knowledge Discovery*, v 5, n 3, pages 213-46, 2001.
5. G. I. Webb, S. Butler and D. Newlands. On detecting differences between groups. In *Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pages 256-65, 2003.
6. S. Brin, R. Motwani and C. Silverstein. Beyond market baskets: generalizing association rules to correlations. In *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*, v 26, n 2, pages 265-76, 1997.
7. S. Brin, R. Motwani, J. D. Ullman and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*, v 26, n 2, pages 255-64, 1997.
8. C. C. Aggarwal, P. S. Yu. A new framework for itemset generation. In *Proc. of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1998)*, pages 18-24, 1998.

9. S. Morishita and J. Sese. Traversing itemset lattices with statistical metric pruning. In *Proc. of the ACM SIGACT-SIGMOD-SIGART Symposium on Database Systems (PODS)*, pages 226-36, 2000.
10. Y. Ohsawa and Y. Nara. Understanding internet users on double helical model of chance-discovery process. In *Proc. of the IEEE Int'l Symposium on Intelligent Control*, pages 844-9, 2002.

An Integrated Approach for Mining Meta-rules^{*}

Feiyue Ye^{1,2}, Jiandong Wang¹, Shiliang Wu², Huiping Chen¹,
Tianqiang Huang¹, and Li Tao¹

¹ College of Information Science and Technology, Nanjing University of Aeronautics and
Astronautics, Postal code 210016, Nanjing, China

{cjsyes8@pub.cz.jsinfo.net}

² Department of Computer Science and Technology, Jiangsu Teachers College of Technology,
Postal code 213001, Changzhou, China

{cjsyes8@pub.cz.jsinfo.net}

Abstract. An integrated approach of mining association rules and meta-rules based on a hyper-structure is put forward. In this approach, time serial databases are partitioned according to time segments, and the total number of scanning database is only twice. In the first time, a set of 1-frequent itemsets and its projection database are formed at every partition. Then every projected database is scanned to construct a hyper-structure. Through mining the hyper-structure, various rules, for example, global association rules, meta-rules, stable association rules and trend rules etc. can be obtained. Compared with existing algorithms for mining association rule, our approach can mine and obtain more useful rules. Compared with existing algorithms for meta-mining or change mining, our approach has higher efficiency. The experimental results show that our approach is very promising.

1 Introduction

Mining association rules is one of the important issues of data mining, and the key of mining association rules is mining frequent patterns. Now Apriori[1] and its enhanced algorithm, FP-growth[2] and CT-ITL[3] algorithm are some important ones on frequent pattern mining in the world. Those algorithms aim at methods and efficiency on mining association rules, but they only fit for mining strong association rules with average support in total. However, the strength of some association rules may change over time. To mining the association rules at changing data sets, some incremental updating algorithm for mining association rule [4] are put forward, but these algorithms mine still rules which are average support and confidence more than or equal to appointed threshold at whole, so they can not use to mine rules which are change over time and predict.

^{*} The work was supported in part by the fund of the Natural Science Plan from University in Jiangsu Province, China, Number: 04KJB460033.

Researchers have put forward some changing mining algorithms [5-9]. Ref. [5] is concerned with a basic framework for mining change from rule sets. To find whether a set of association rules discovered in a time period is applicable in other time periods was discussed in [7]. An approach to find a decision tree change between two time periods was proposed in [6]. Ref. [9] is concerned with meta-mining. The method of mining changes in association rules using fuzzy decision trees was put forward in [8]. In these algorithms, only the algorithms that were mentioned in [5][7][8][9] can be used to mining change of association rule. However, these algorithms only consider the mining change of association rule based on rule sets that have been mined by some of algorithms for mining association rule, so they do not consider the integrated efficiency for the whole mining process.

This paper presents an integrated approach of mining association rules and meta-rules based on a hyper-structure, this approach is evidently different from the above mentioned algorithm. With this approach, we can mine various association rules and meta-rules, for example, stable association rules, trend association rules, etc. In this paper, a classification approach, based on neural network, to classify association rule sets is discussed and the corresponding experiment is performed.

2 Constructing Hyper-Structure

In this section, Hyper-Structure is constructed. We first define the problem of 1-frequent itemset projected database.

Definition 1. Let $I = \{i_1, i_2, \dots, i_m\}$ be a set of all item in transaction database D_0 that there are N transactions, $X_{k'}$ is the transaction itemsets of the k' -th transaction, $X_{k'} \subseteq I$, i.e., $X_{k'} = \{i_{k_1}, i_{k_2}, \dots, i_{k_n}\}$, where $1 \leq j < n, 1 \leq k_j < k_{j+1} \leq m$. $X_{(n)}$ denotes that the set X contains n items. The number of transactions in D_0 containing itemset X is called the support of X , denoted as $sup_0(X)$. Given a minimum support threshold s , if $sup_0(X) \geq s$, then X is frequent in D_0 . Let i'_{p_j} be a frequent item in D_0 , called as 1-frequent item, where $1 \leq p_j \leq m$, and Let $I' = \{i'_{p_1}, i'_{p_2}, \dots, i'_{p_m}\}$ be the set of all 1-frequent item, where $1 \leq m' \leq m$ and $I' \subseteq I$. Thus the projection between I' and $X_{k'}$ is $A_{k'}$, and $A_{k'} = I' \cup X_{k'} = \{i'_{q_1}, i'_{q_2}, \dots, i'_{q_n}\}$, and then the transaction database that consists of A_1, A_2, \dots, A_N is called 1-frequent itemset projected database A .

2.1 Structure of Hyper-Structure Head Table

The hyper-structure head table contains two fields: item number field and pointer field. The pointer in pointer field points to a hash chain structure with the same number of items. The hyper-structure head table is created dynamically. The hyper-structure is illustrated in Fig.1.

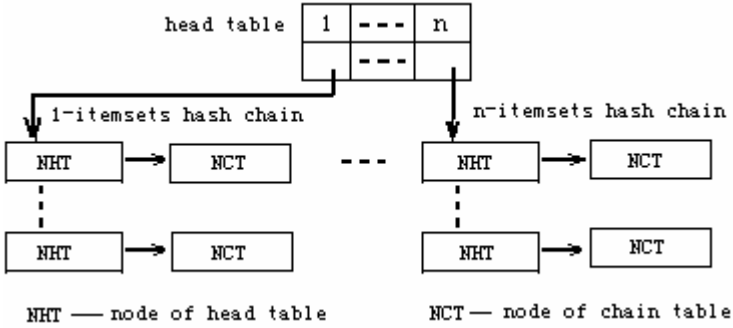


Fig. 1. Hyper-structure

2.2 Chain Address Function

The hash function of the item i_{k_j} (where k_j is the item number) in 1-frequent itemsets is given below:

$$h(k_j) = k_j \tag{1}$$

Let $B = \{q_1, q_2, \dots, q_{n'}\}$ be a set of the item number in itemset $A_{k'} = \{i_{q_1}^{k'}, i_{q_2}^{k'}, \dots, i_{q_{n'}}^{k'}\}$. The hash function of the multi-item itemsets is given below:

$$h(q_1, q_2, \dots, q_{n'}) = \left(\sum_{i=q_1}^{q_{n'}} (2i+1)z_i \right) \text{mod } p' \tag{2}$$

If $i \in B$, then $z_i = 1$, otherwise $z_i = 0$; p' is the sum of the adjusted pattern of the multi-itemsets.

2.3 Chain Address Structure

The node structure of the heat table and the chain table of 1-frequent itemsets is illustrated in Fig.2 and Fig.3 respectively.

The chain address is produced according to formula (1) in Fig.2. The pointer points to the node structure of chain table.

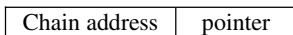


Fig. 2. The node structure of head table

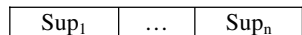


Fig. 3. The node structure of chain table

The node structure of head table of multi-itemsets is shown in Fig.4 and the node structure of the chain table of the multi-itemsets is shown in Fig.5:

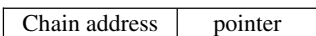


Fig. 4. The node structure of head table

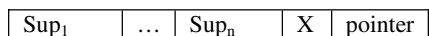


Fig. 5. The node structure of chain table

The chain address is obtained from formula (2) in Fig.4 ,Here, “pointer” points to its chain table node; “ Sup_i ” is used to record the count of itemsets X which appears in D_i ;The pointer points to the next chain table node in Fig.5.

2.4 The Algorithm of Constructing Hyper-Structure

Algorithm 1: the algorithm of constructing the hyper-structure

Input: A transaction database D_0

Output: Hyper-structure

Method:

1. scan transaction database D_0 to obtain the set of 1-frequent item and its projected database and maximal projected item $Max(|X|)$, and partition the projected database A of D_0 into $D_1^A, D_2^A, \dots, D_n^A$ according to time period t_1, t_2, \dots, t_n ;
2. construct the head table of hyper-structure from 1-item to $Max(|X|)$;
3. construct 1-frequent item hash chain;
4. *for* ($i=1$; n ; $i++$) {
 $j=1$;
do while the scan for projected database D_i^A is unfinished
 forall join the item in item number set B of A_j to generate itemset X *do* {
 calculate value of $h(q_1, q_2, \dots, q_{n'})$ for X ;
 locate address of $h(q_1, q_2, \dots, q_{n'})$ in $|X|$ -item hash chain;
 if there has not been X in corresponding address chain table in
 $|X|$ -item hash chain according to the value of $h(q_1, q_2, \dots, q_{n'})$ *then*
 {save X to there and $sup_i = 1$; }
 else { $sup_i = sup_i + 1$; } }
 $j=j+1$;}

3 Association Rule and Meta-rule Mining

The association rule mining problem can be decomposed into two subproblems (Agrawal et al., 1993), That is, Finding all frequent itemsets and using the frequent itemsets to generate the desired rules. The meta-rules mining problem is to find the change of the association rule over time. So the main mission of mining association rules and meta-rule is mining frequent itemsets. Having got a frequent itemset, we can further obtain all its subsets, and form the corresponding rule set. To mine the association rules and meta-rules from the hyper-structure, the association rule and the meta rule are defined below:

Definition 2. An association rule r is denoted by “ $X_1 \Rightarrow X_2$ with caveat c ”, where $X_1 \subset X$ and $X_2 \subset X$, and $X_1 \cap X_2 = \phi$, c is support and confidence.

Definition 3. Let min_sup be a minimum support threshold, min_conf be a minimum confidence threshold, if rule r satisfies both $support(X_1 \Rightarrow X_2) \geq min_sup$ and $confidence(X_1 \Rightarrow X_2) \geq min_conf$, then r is called strong rule. R denotes the set of rule r .

Lemma 1: If itemsets X is global frequent, then it is frequent at least in one segment D_i ($1 \leq i \leq n$).

When mining the meta-rule, we consider only the rules which are global frequent and are frequent in one time segment, According to **Lemma 1** the meta-rule is defined in the following:

Definition 4. Let the support and confidence of rule $X_1 \Rightarrow X_2$ from datasets $D_0, D_1, D_2, \dots, D_n$ be $sup_0, sup_1, sup_2, \dots, sup_n$ and $conf_0, conf_1, conf_2, \dots, conf_n$ respectively, if $sup_0 \geq min_sup$ and $conf_0 \geq min_conf$, then the support meta-rule m_s from D_1, D_2, \dots, D_n is given below:

$$X_1 \Rightarrow X_2 : \{sup_1, sup_2, \dots, sup_n\},$$

And the confidence meta-rule m_c from D_1, D_2, \dots, D_n is given below:

$$X_1 \Rightarrow X_2 : \{conf_1, conf_2, \dots, conf_n\}.$$

Lemma 2: All subsets of the frequent itemsets that appear in n -itemsets hash chain must also appear in corresponding 1-itemsets to $(n-1)$ -itemsets hash chain.

Apriori property: All nonempty subsets of a frequent itemset must also be frequent. According to above definition and lemma and property, the algorithm for mining association rules and meta-rules is given in the following:

Algorithm 2: The algorithm for mining association rules and meta-rules

Input: hyper-structure; dataset D_0 and dataset D_1, D_2, \dots, D_n ; support threshold min_sup_i ($i = 0, 1, \dots, n$) and confidence threshold min_conf_i ($i = 0, 1, \dots, n$)

Output: Association rules and meta-rules

Method:

Association rule set $R = \phi$; support meta-rule set $M_s = \phi$; confidence meta-rule set $M_c = \phi$;

For ($i=2;n;i++$) {

The pointer points to the first address of i -items hash chain;

Do while to search i -item hash chain is unfinished {

Search the chain table node at the address and obtain itemset X and $sup_1, sup_2, \dots, sup_n$ and sup_0 ;


```

If  $sup_0 \geq min\_sup_0$  then {
  If  $i > 2$  then {
    Forall the subsets  $X_{(i-1)}$  and  $X_{(i-2)}$  of the itemset  $X$  do {
      Forall the itemsets  $X_{(i-1)}$ ,  $X_{(i-2)}$ ,  $X_{(i-2)}$  and  $X_{(i-2)}$  to satisfy definition 2 are
        regarded as antecedent and consequent for forming association rules do
          {
            Obtain the  $sup_j(j=0,1,\dots,n)$  of antecedent and consequent of association
              rule respectively;
            Calculate  $conf_j(j=0,1,\dots,n)$  according to  $sup_j(j=0,1,\dots,n)$ ;
            If  $conf_0 \geq min\_conf$  then add the rule to  $R$  and add support meta-rule
               $m_s : (sup_1, sup_2, \dots, sup_n)$  to set  $M_s$  and add confidence meta-rule
               $m_c : (sup_1, sup_2, \dots, sup_n)$  to set  $M_c$ ; } }
          else {
            search and obtain the corresponding  $sup_j(j=0,1,\dots,n)$  of two item of  $X$ ;
            Calculate  $conf_j(j=0,1,\dots,n)$  according to  $sup_j(j=0,1,\dots,n)$ ;
            If  $conf_0 \geq min\_conf$  then add the rule to rule sets and add support
              meta-rule  $r : (sup_1, sup_2, \dots, sup_n)$  to meta-rule set  $M_s$  and add confidence
              meta-rule  $r : (sup_1, sup_2, \dots, sup_n)$  to meta-rule set  $M_c$ ; } }
          else {stop mining} }

```

4 Analysis of the Change Trend of Association Rules Using Meta-rules

The types of change trend of association rules can be divided into several cases in the following:

Stable rules: These rules do not change a great deal over time. Stable rules are more reliable and can be trusted.

Trend rules: These rules indicate some underlying systematic trends.

Irregular or random movements: These characterize the sporadic motion of time series due to random or chance events.

Cyclic movement or cyclic variation: These refer to the cycles, that is, the long-term oscillations about a trend line or curve, which may or may not be periodic.

Seasonal movements or seasonal variations: These movements are due to events that recur annually.

In this paper, only forefront three types are considered.

Apparently, if the meta-rules are directly analyzed using the trend analysis method, each rule will be scanned and calculated repeatedly. It is necessary to first classify the

meta-rules in order to improve the analysis efficiency. That is, the meta-rules are classified into several classes which donate a change trend of association rules. For example, the association degree is stable or decreased or increased over time, etc. Then each class meta-rules are respectively analyzed on demand. There are many existing classification methods, for example, the SVM method, neural network method, C4.5 algorithm, Bayesian classification, etc. Next we are going to discuss classification method for meta-rules using BP neural network.

Before using neural network for classification, some training samples need to be obtained. These samples should be formed according to the following cases:

- 1) The association degree of rule sets is stable movement over time, its output is y_1 .
- 2) The association degree of rule sets is increased over time, its output is y_2 .
- 3) The association degree of rule sets is decreased over time, its output is y_3 .
- 4) The association degree of rule sets is random movements over time, its output is y_4 .

First, the BP network is trained by sample data sets to obtain its weight, and then trained BP network is applied to corresponding classification work. In the process of mining the meta-rule, the meta-rule which has already mined from hyper-structures will be imported into the BP network defined before so that the classified association rules can be obtained.

Generally speaking, after classification the usability of association rules is improved considerably, and they can be further analyzed expediently.

5 Experimental Result

In our experiments, the dataset is from a supermarket transaction database from October 1 in 1996 to May 31 in 1997, its original size is 50.6M, its worked size is 11.6M, and let it be D_0 , there are 47536 transactions in D_0 . We partition D_0 into D_1, D_2, D_3, D_4 according to time period from t_1 to t_4 , and the partitioned result is shown in Table 3.

Table 3. Partitioned result

Sub-database	D_1	D_2	D_3	D_4
Time period	t_1 (96.10-96.11)	t_2 (96.12-97.01)	t_3 (97.02-97.03)	t_4 (97.04-97.05)
The number of transaction	10918	10456	13801	12361

By setting minimum support to 0.04% and minimum confidence to 50%, the association rule set R_0 and corresponding meta-rule set M_s and M_c are discovered in the hyper-structure that is constructed through scanning the sub-database D_1, D_2, D_3, D_4

respectively. There are 25 entry association rules in R_0 . There are 25 entry support meta-rules and 25 entry confidence meta-rules in M_s and M_c respectively. The classified result of the support meta-rule set is showed in Table 4. The classified result of the confidence meta-rule set is omitted.

Table 4. Classified result

Type of rules	Stable	Increased	Decreased	Random
Number of rules	3	0	3	19

By observing the classified result we can discover that classified result is correct. The supports of 3 entries “Stable” rules are all more than support threshold from t_1 to t_4 and support change is little. The support of 3 entries “Decreased” rule falls gradually from t_1 to t_3 , and the rules are perished in t_4 . In “Random” column, except 1 entry rule appears both t_1 and t_3 , the rest only appear in one time period. So the association rules obtained from R_0 is not all usable and trend analysis is necessary. Experimental result indicates that the usability of association rules obtained by our approach is improved considerably.

6 Discussion

Existing algorithm for mining association rule [1-4] can only mine strong association rules with average support and confidence in total. Our approach is distinctly different from above mentioned algorithm, as not only it can mine strong association rules with average support and confidence in total, but also can mine more various association rules. Experimental result appears that our approach can mine the association rules that have better usability.

Compared with existing algorithm of meta-mining or change mining [8-9], our approach has higher efficiency. In existing algorithms of meta-mining or change mining, first the data sets are partitioned into several subset according to time segments, and then each subset is mined by existing algorithm for mining association rule to gain association rule sets, finally, the meta-rule sets or trend rules are mined from association rule sets. With the approach proposed in this paper, global strong rule sets, meta-rule sets and the classification of the meta-rule sets can be obtained by scanning database only twice, so our approach has lower I/O spending than above mentioned algorithm.

In addition, our algorithm will generate less numbers of 2-itemsets than Apriori algorithm, because the 2-itemsets is generated by linking all 1-frequent item in Apriori algorithm. However, in our algorithm, the 2-itemsets is generated by linking 1-frequent item in intersection between 1-frequent itemset and each transaction. In our algorithm, the confidence of association rule can be calculated by obtaining the count of corresponding itemsets from the hyper-structure directly.

7 Conclusion

In this paper, we put forward an integrated approach for mining association rules and meta-rules. With the approach, the association rules and meta-rules can be mined for time serial database effectively, Not only has it high efficiency, but also has more powerful mining capability, It has some distinct advantages compared with existing algorithms. A formalized expression of meta-rules for time serial databases is given, thus meta-rules can be denoted and processed expediently, this offer a new pass for expression or re-mining of meta-rules.

References

- [1] Agrawal R. and Srikant R.: Fast algorithms for mining association rules. In VLDB'94(1994), 487-499.
- [2] Han J., Pei J. and Y.Yin.: Mining frequent patterns without candidate generation. In SIGMOD'00(2000), 1-12
- [3] Yudho G.S., Raj P., Gopalan.: CT-ITL: Efficient Frequent Item Set Mining Using a compressed Prefix Tree with Pattern Growth. 14th Australasian Database Conference(ADC2003)(2003).
- [4] Yang M., Sun Z.H., Song Y.Q.: Fast Updating of Globally Frequent Itemsets. Journal of Software,(8)(2004)1189-1196.
- [5] Spiliopoulou M., Roddick J.F.: Higher order mining: modelling and mining the results of knowledge discovery, Conf. on Data Mining Methods and Databases for engineering, Finance, and Other Fields, WIT Press, Southampton, UK(2000)309–320.
- [6] Liu B., Wynne H., Heng S.H. *et al.*: Mining Changes for Real-life Applications, in The 2nd International Conference on Data Warehousing and Knowledge Discovery, UK(2000).
- [7] Bing L., Wynne H. and Ming Y.: Discovering the Set of Fundamental Rule Changes, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2001).
- [8] Wai H. A., Keith C.C.: Mining changes in association rules: a fuzzy approach. Fuzzy Sets and Systems, In Press, Corrected Proof, Available online 11 September 2004(2004).
- [9] Abraham T., Roddick J. F.: Incremental Meta-mining from Large Temporal Data Sets, Advances in Database Technologies, Proceedings of the 1st International Workshop on Data Warehousing and Data Mining (DWDW'98)(1999),41-54.

Data Mining on Crash Simulation Data

Annette Kuhlmann¹, Ralf-Michael Vetter¹, Christoph Lübbing²,
and Clemens-August Thole¹

¹ Fraunhofer Institute for Algorithms and Scientific Computing (SCAI),
Schloss Birlinghoven, 53754 Sankt Augustin, Germany

{kuhlmann, vetter, thole}@scai.fraunhofer.de

² BMW AG, EK-210, Knorrstrasse 147, 80788 München, Germany
christoph.luebbing@bmw.de

Abstract. The work presented in this paper is part of the cooperative research project AUTO–OPT carried out by twelve partners from the automotive industries. One major work package concerns the application of data mining methods in the area of automotive design. Suitable methods for data preparation and data analysis are developed. The objective of the work is the re–use of data stored in the crash–simulation department at BMW in order to gain deeper insight into the interrelations between the geometric variations of the car during its design and its performance in crash testing. In this paper a method for data analysis of finite element models and results from crash simulation is proposed and application to recent data from the industrial partner BMW is demonstrated. All necessary steps from data pre–processing to re–integration into the working environment of the engineer are covered.

1 Introduction

The objective of the data mining work presented in this paper is the re–use of data stored in the crash–simulation department at BMW in order to gain deeper insight into the interrelations. Here the objective is to find hidden knowledge in stored data. In principle one could think of various possible questions for such a knowledge mining analysis:

- which innovations have evolved during the design process
- were certain steps in the development unnecessary or could they be shortened
- is it possible to extract analogies between different car projects
- can reasons that have lead to certain design decisions be reproduced
- can this reasoning be applied to future projects

The data mining project in AUTO–OPT aims at examining the applicability of data mining methods on crash simulation data [1]. Due to the fact that design and development knowledge is the major asset of engineering, an automotive company cannot be expected to share large amounts of their data for research reasons. On the other hand, interesting results from data mining can only be achieved from interesting data. Therefore in this work the applicability of the

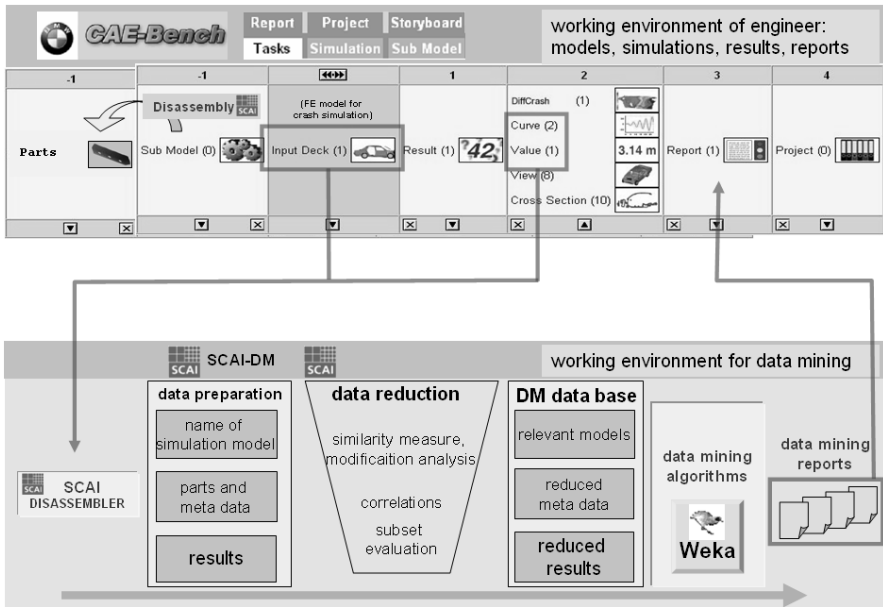


Fig. 1. Procedure for data mining. Models and crash results are exported from CAE-Bench, models are disassembled into parts, geometry based meta data are calculated, parts and meta data are stored within SCAI-DM, crash results are attached to meta data, data mining tables are assembled, DM analysis is performed, result files are produced and exported

method is demonstrated, its value cannot be evaluated on the data basis available. This will be aimed at in future work.

The crash department at BMW stores all relevant information in the simulation data management system CAE-Bench [2]. Data mining queries are to be submitted from this environment. Results have to be brought back into this system and assigned to the underlying models, i.e. stored within their audit trail. This procedure is schematically shown in Figure 1.

Fraunhofer SCAI has been provided with data from one of the most recent car projects at BMW. The vehicle under development is shown in Figure 2 (left). Each data set describes one stage of construction of this car within the development process via finite element models (FE models) made up by about 500.000 independent nodes and elements.

Each car is composed of app. 1200 parts. CAE-Bench stores the models as complete vehicles, i.e. one single large FE model, called input deck. In order to analyse the geometry of the parts these input decks need to be disassembled as shown in Fig. 2 (right) for an older BMW vehicle, the new model cannot be shown in such detail because of a nondisclosure agreement.



Fig. 2. Recent model of BMW employed for data mining (left). One FE input deck consisting of numerous parts (right)

2 Preparation of the Data for Data Mining

It is generally accepted that the preparation of the data involves as much as 80-90% of the effort when a data mining task is attempted, see e.g. [3]. The data cannot be processed by a data mining tool in their original format. To the authors best knowledge no approach for data mining on raw finite element data exists. The preparation of the data thus constitutes the main challenge for the data mining approach on the FE data. In addition, the data has to be cleaned and checked for consistency and the appropriate values have to be combined. As a first but major step a process for data preparation has been developed:

- a) Export of data from CAE-Bench
- b) Disassembling into parts and computation of meta data
- c) Data cleaning and sorting — clustering of parts
- d) Similarity analysis and data reduction — clustering of variants
- e) Evaluation and cleaning of crash result data

As a result of this procedure a table is generated that allows for access to the data with data mining algorithms. This section focuses on the preparation of the data, whereas the application of the data mining algorithms will be presented in section 3.

a) Export of Data from CAE-Bench

CAE-Bench can export selected input decks along with the result achieved when these models were subjected to a virtual crash test. An example is shown in Fig. 4. Information is extracted from this export, such that the relevant crash results can be attached to the respective input deck data and stored in the SCAI data mining framework (SCAI-DM).

b) Disassembling into Parts and Computation of Meta Data

Motivation. The data mining approach in this work concerns the shape of the parts of the car. The aim is to analyse how changes in shape have influenced crash behaviour. The FE-model itself contains all geometrical information. However, this information is hidden from data mining algorithms, as these cannot extract

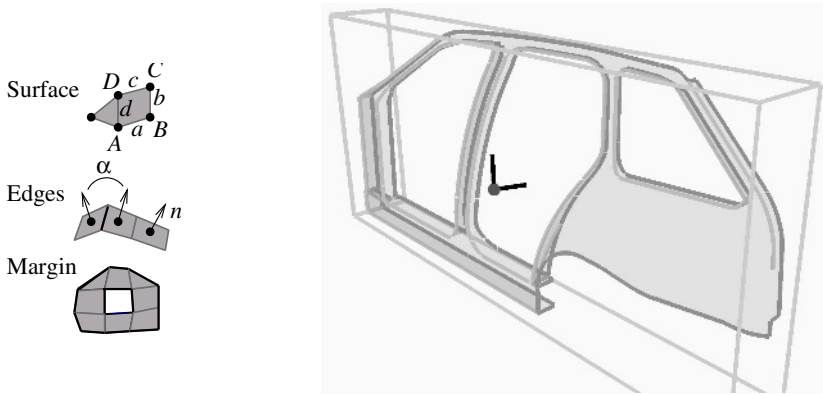


Fig. 3. Typical meta data and their appearance in an example part

meaningful knowledge from node and element descriptions. Meta data has to be determined such that it quantifies geometry in an appropriate manner. In this work several values have been chosen as meta data, e.g. the centre of gravity of each part, the moments of inertia, the length of edges and margins, surface size, bounding box, length of branching lines—as shown in Figure 3. All of these are mesh independent. They thus enable comparison between models that have been meshed with different algorithms or programs. The meta data reduce the amount of data massively, such that handling of data is facilitated considerably.

Reading the Input Deck. Today the body-shell of a finite element car model is described by an input deck of 100 MB containing approximately 1.500.000 lines. Figure 4 shows a small subset of such an input deck. The part number indicates which element of the meshes belongs to which specific part of the car. The material section defines a homogeneous density and thickness for each part. In the disassembling procedure all elements with the same part number and their respective nodes are extracted from the input deck and form one new mesh for this single part.

For each part the disassembler thus extracts a sub-mesh of the input deck. This sub-mesh is the basis of the calculation of the meta data. The sub-mesh files are also used to create previews of the parts: from three different angles or—on demand—in form of a three-dimensional applet visualisation [4, 5, 6]. Since the generation of previews of parts is a time intensive process, it is initiated only for new parts which have not previously been stored in the database.

Computation of Meta Data. For meta data calculation various details on FE models have to be taken into account. The model surfaces here are curved. Using shell elements for the description means that the four corner points of a quadrilateral do not necessarily lie in one common plain, see Fig 3. One well defined way to calculate their surface is $S \approx \frac{1}{2} [(\mathbf{a} + \mathbf{c}) \times (\mathbf{b} + \mathbf{d})]$. The mass of an element is given by its surface multiplied by the material thickness d and density ρ . The centre of gravity, at which the mass m is assumed to be located

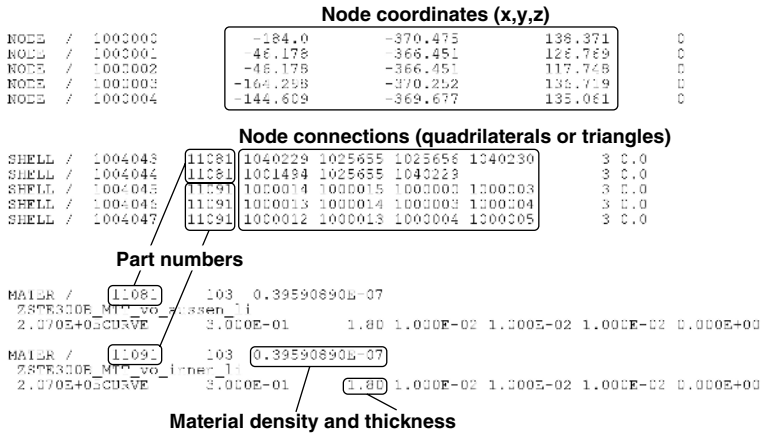


Fig. 4. Excerpt of an input deck: the NODE and SHELL sections describe the geometry of the finite element meshes; in the MATER section the thickness d and density ρ of the material can be found. Beside shell elements, which can be triangles or quadrilaterals, additional elements like membrane elements (with 4 nodes), solid (8 nodes), beams (2 nodes), or bars (3 nodes) appear in an input deck

in a single element is positioned approximately at $\frac{1}{4}[\mathbf{A} + \mathbf{B} + \mathbf{C} + \mathbf{D}]$, where $\mathbf{A} \dots \mathbf{D}$ are the corner points of the SHELL element. Then the centre of gravity and the moments of inertia of the complete part are given by their sum over all point masses. For every element a normal vector is constructed by $\mathbf{n} = \frac{\mathbf{a}+\mathbf{c}}{|\mathbf{a}+\mathbf{c}|} \times \frac{\mathbf{b}+\mathbf{d}}{|\mathbf{b}+\mathbf{d}|}$. The normal vectors \mathbf{n}_1 and \mathbf{n}_2 of adjacent elements are used for detecting edges. If the angle $\alpha = 2 \arcsin(|\mathbf{n}_2 - \mathbf{n}_1|/2)$ is larger than a user defined value, the connection line between the elements is called an edge of the mesh. If one side of an element is not connected to any further element, this line is assumed to be a margin of the structure. In this manner all meta data characterising the geometry of each part is computed.

c) Data Cleaning and Sorting — Clustering of Parts

The finite element models are subject to numerous kinds of modifications. During the engineering process in which a car model is improved with respect to its crash-worthiness a subset of parts is modified. In general the parts modified are the crash-relevant ones. Additional modifications follow the demands of other engineering disciplines, e.g. holes may be inserted into the parts in order to achieve a better drain of varnish during production. Such measures reduce crash-worthiness, which then again has to be improved by further modifications.

However, not all parts are modified in all stages of car design. As unchanged parts cannot be responsible for deviations in the simulation results, such parts can be excluded from the analysis. In order to remove the parts that never have been modified MD5 checksums are created for all sub-mesh files, Fig. 5 (right column). If the checksum of any part stays constant in any data set of interest









		Model	Number	Mass	Margin	Surface	Edges	MD5Sum
side view 	top view 	E6016VS01_fd01v02	11171	3.72113E-4	2323.04	49221.4	473.896	ed6bee68c9d11dcfb09d12839d
Framehofer SCAI								
side view 	top view 	e60ppp_fd08v11a	11171	5.42834E-4	2009.7	52771.9	480.21	c261dc25558729900093c57f25d
Framehofer SCAI								
side view 	top view 	E60PPP_fd01v01a	11171	5.89499E-4	1469.6	57328.8	757.541	4d5647dbd0ceedb6b213ba89df
Framehofer SCAI								
side view 	top view 	e60ppp_fd01v01b	11171	5.89499E-4	1469.6	57328.8	757.541	4d5647dbd0ceedb6b213ba89df
Framehofer SCAI								

Fig. 5. Screenshot of SCAI-DM framework, here some variations of part no. 11171 with meta data and checksums. The database contains about 146.000 parts belonging to 134 different crash tests. Deleting all sub-mesh files with multiple MD5 checksums (right column) the database can be reduced by 93% to 9900 different parts

the part was left unchanged and one single reference of the sub-mesh file is stored. Solely parts with more than one instance in the data base are included in the data mining queries.

SCAI-DM Data Base. After disassembling all parts are stored in the SCAI-DM framework along with their meta data, as shown in Fig. 5. Depending on the purpose parts and data can be displayed in any other combination.

Avoiding Inconsistent Naming/Numbering. One bottle neck for data mining of the BMW data is the fact that text entries in the data management system are free text. Some agreements are complied with in the majority of cases. Repeatedly, however, re-naming and re-numbering of parts was encountered in the data, which showed that rules were not consequently followed. Therefore, to avoid irrelevant results from the analysis aimed at it is vital that all data entering the analysis stick to the same rules. The safest way to achieve correct data is to avoid the text entries in CAE-Bench altogether and use the FE descriptions as a basis. This again implies that an automatic method to identify parts has to be set up such that the use of part-numbers or -names coming from CAE-Bench is avoided. The meta data calculated from the FE model can be the basis for part identification using cluster analysis. The clustering process divides a dataset into mutually exclusive groups such that the members of each group are as "close" as possible to one another, and different groups are as "far" as possible from

one another, where distance is measured with respect to all available variables, see e.g. [7].

Each meta data property spans a new dimension in the similarity space. The meta data of a specific part are represented by a point in the multidimensional similarity space. Similar parts have similar meta data and form a cloud of adjacent points. Figure 6 (left) shows the idea of clustering of meta data in a schematic diagram: Two dimensions of the similarity space are shown. The meta data of the parts L and C form two clouds, in which a substructure indicates the presence of several modifications. Figure 6 (right) shows a clustering plot of the BMW data. The dots describe to two different parts with the same part number 11011 "Motorträger" (three clusters on the right) and "Schottblech Motorträger" (small cluster in the centre of the diagram). This is an example for a change in numbering of a part.

d) Similarity Analysis and Data Reduction — Clustering of Variants

Differing checksums indicate that a sub-mesh was modified in some unknown way. Then negligible file modifications have to be distinguished from relevant changes such as modified shapes. In this framework for data mining the geometric meta data, as described above, serves as a similarity measure for the parts. Minor and major changes of the parts design will result in a hierarchical structure of clouds and sub-clouds, see Fig. 6 (left). Using hierarchical clustering a substructure can be found inside the clusters. Starting with C_2 as a reference, C_1 contains parts with a higher mass (caused by a higher thickness d or density ρ of the material) while the parts in C_3 result from geometrical modifications increasing the surface (e.g. caused by additional beadings for higher stiffness). In the clustering plot of BMW data, Fig. 6 (right), the light grey dots belong to three modifications of the same part, namely 11011 "Motorträger". An example for typical modifications and their influences on the meta data can be seen in Fig. 5, where similar parts have been selected from the data base.

This clustering of parts in the meta data space in order to identify variants of designs is a time consuming task when all relevant parts and meta data are

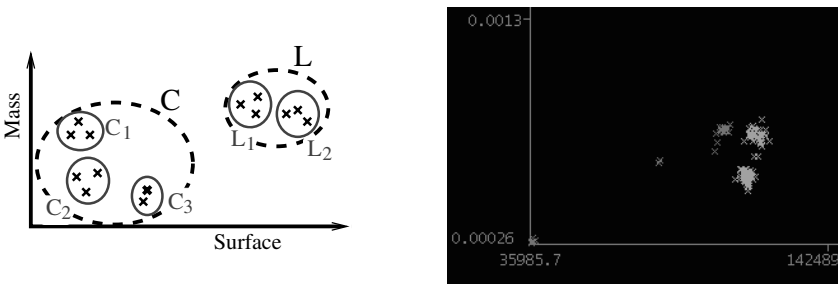


Fig. 6. Left: idea of hierarchical clustering in a schematic diagram. Two dimensions of the similarity space are shown. Right: meta data of real parts. The grey clusters on the right correspond to three different variants of the same part

considered. An alternative method leading to similar results is to merge the meta data into a single similarity measure [8]. For the work presented in this paper a weighted sum of all the meta data has been employed. Then, if the weights are appropriately chosen, parts with the same similarity measure are similar in shape. This similarity measure serves as the main attribute for data mining, as described in section 3.

e) Evaluation and Cleaning of Crash Result Data

For each crash simulation several values and curves, as well as images and movies are computed in order to evaluate the crash worthiness of this particular design. The bottle neck here is similar as before: the scripts that calculate the values stored in CAE-Bench can be altered at any time, such that the compatibility of the values has to be ensured before data mining can be attempted. No automatic approach could be developed to check this compatibility so far. In this work values whose scripts have been left unchanged for all simulations have been used for the DM analysis. This could, however, be a serious drawback of the method and other possibilities of ensuring reproducible values for the crash results have been discussed with BMW. In this paper the only result values analysed are intrusions. Intrusions measure the difference between the distances of two points inside the car (one FE node) before and after the crash test.

In the last step of data preparation the data base is reordered. A table containing one line per crash test is formed: the name of the model, the similarity values of the parts and the result values of interest.

3 Datamining on Similarity Data

The aim of this work is to evaluate the applicability of data mining methods for simulation data in engineering. As a result from a complex data preparation procedure a table suitable for data mining can be achieved in which simulation data appears transformed into geometrical meta data. This table (Fig. 7) is written in Weka format, for which readily applicable data mining algorithms are available, see [11, 12].

3.1 Attribute Selection

An important step in data mining is the selection of those attributes that are relevant predictors before starting to build the model [9]. This is important because too many may be available when the full data set is encountered. Irrelevant information should be excluded from the data set [10]. Thus a feature selection algorithm can show which attributes have the strongest influence on the class. For the crash simulation data this information can be particularly valuable, as it reduces a vast amount of geometrical modifications to a small number of seemingly important ones.

Employing an attribute selection algorithm on the crash data, e.g. ChiSquared in Weka, means that parts are ranked depending on the impact of the variation

Model	Sim_8010	Sim_11011	Sim_11012	Sim_11013	Sim_11014	Sim_11021	Sim_11022	...	Intrusion 1	Intrusion 2
E6016V501_fd01v02	0.13	3.11	3.11	0.91	0.91	0.53	0.53	...	-252.578	-217.411
E6016V501_fd01v02a	0.13	3.11	3.11	0.91	0.91	0.53	0.53	...	-248.203	-217.2
e60vbg1+_fd02-1v06	0.0	1.13	1.0	4.08	4.02	0.01	0.01	...	-110.295	-114.728
e60vbg1+_fd02:redv05	0.0	1.29	1.0	4.05	4.02	0.01	0.01	...	-191.374	-180.895
e60vbg1+_fd02v04	0.0	1.29	1.0	4.05	4.02	0.01	0.01	...	-118.985	-137.677
e60vbg1+_fd02v04b	0.0	1.29	1.0	4.05	4.02	0.01	0.01	...	-116.046	-126.97
e60vbg1+_fd02v07	0.0	1.13	1.0	4.08	4.02	0.01	0.01	...	-117.73	-123.492
e60vbg1+_fd03-1v09	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-155.667	-169.425
e60vbg1+_fd03-2v10	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-139.427	-145.34
e60vbg1+_fd03-2v10a	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-148.988	-163.872
e60vbg1+_fd03-2v10b	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-156.212	-170.325
e60vbg1+_fd03-3av12	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-148.141	-146.928
e60vbg1+_fd03-3bv12b	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-142.861	-162.149
e60vbg1+_fd03-3v11	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-146.973	-153.957
e60vbg1+_fd03-4v13	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-142.953	-145.71
e60vbg1+_fd03-5v02	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-158.559	-166.717
e60vbg1+_fd03v08	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-117.151	-126.494
e60vbg1+_fd04v14a	0.0	1.37	1.37	3.53	3.54	0.35	0.35	...	-148.246	-173.806

Fig. 7. Re-ordered table for data mining: each line contains one model with similarity values of relevant parts and serves as instance for analysis. Simulation results (intrusions) are attached and serve as classes

Attribute Evaluator (supervised, Class (nominal): 351 value_Some_Intrusion):		
Chi-squared Ranking Filter		
Ranked attributes:		
33.5022	1 Sim_11013	H1A_MOTORTRAEGER VO_1
28.02819	5 Sim_15121	G1A_SCHOTTBLECH EINSTIEG HI_1
21.1853	148 Sim_11032	I1A_BUCHSE VO MOTORTRAEGER VO_2
21.1853	146 Sim_11042	I1A_BUCHSE HI MOTORTRAEGER VO_2
21.1853	147 Sim_11041	I1A_BUCHSE HI MOTORTRAEGER VO_1
21.1853	149 Sim_11031	I1A_BUCHSE VO MOTORTRAEGER VO_1

Fig. 8. Attribute selection with Weka: The six parts whose variations have most influence on the intrusion. BMW has confirmed the importance of these parts for the front crash simulated here

of their similarity measure on the intrusion of interest. In Figure 8 one result of such a calculation is shown. The list of parts shows those 6 out of 1200 whose variations have most influence on the intrusion. In this case the data basis is 30 models—the portability therefore is likely to be rather limited.

3.2 Decision Trees

Another method employed in order to demonstrate the possible outcome of data mining on simulation data is the decision tree method. Here a further step has been taken towards the achievement of results relevant to the application engineer. In practice the engineer is very rarely interested in the behaviour of only one of his result values, instead he needs to get an understanding of the influences of his design modification on a range on values. For this reason four result values were selected and clustered into three groups, one of which covers the most desired vehicle behaviour during crash. The clustering of the instances into three groups is demonstrated in Figure 9 for two of the result values. A clear grouping into "good" (circles), "medium" (squares) and "poor" (triangles) crash tests can be seen.

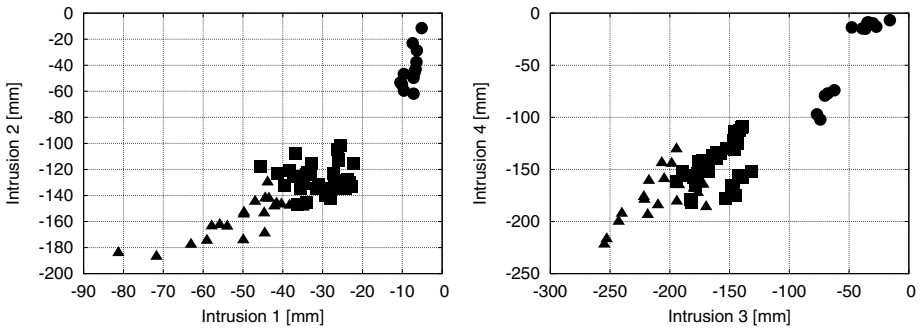


Fig. 9. Clustering of result values "Intrusion 1 ... 4" in order to be able to represent various aspects of crash behaviour with one single class value

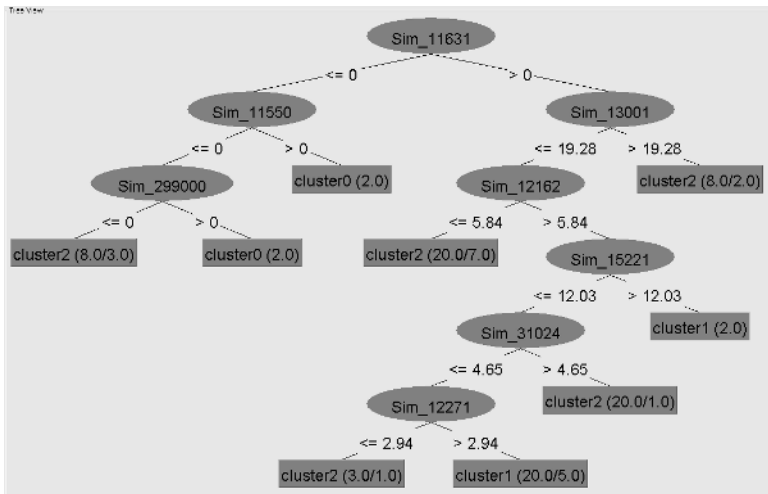


Fig. 10. Decision tree `weka.classifiers.trees.J48 -C 0.25 -M 2`. The existence of the part 11631 "Abstützung Lenksäule Unterteil" is the determining factor for the intrusions. If this part was integrated (> 0) further important parts in this data basis are two modifications of part 13001 "Bodenblech vorne" and part 12162 "Verbindung Längsträger"

The membership of a model to these clusters is then used as "class" when the decision tree is built. As attributes the similarity measure—in this case again a weighted sum of all meta data—is employed.

An example for such a tree is shown in Figure 10. The tree thus shows in which cluster a carmodel can be expected to lie depending on the geometrical version of the parts contained in the carmodel. These represent now the nodes of the tree.

For this example a data basis of 77 crashtests has been employed, which still is a rather small basis for rule building. However, these results are promising

because the similarity measure seems capable of adequately representing shape modifications and lead to meaningful results.

3.3 DM Reports

The results achieved within the SCAI-DM framework is imported into CAE-Bench in order to be accessible by other engineers at other times. The reporting tool in CAE-Bench can include text and figures, such that a data mining report can be stored with the underlying input decks in CAE-Bench. This closes the circle of the procedure shown in Fig. 1.

4 Results

The applicability of data mining on crash simulation data has been demonstrated in this work. A framework for data preparation has been developed. The computation and handling of meta data for similarity search has been studied in detail. The employed similarity measure has proved to be appropriate for detection of relevant changes in shape. The usability of the approach on data from an automotive application has been shown. Due to the limited amount of data available for this work conclusions are limited, but first significant results have been achieved on a test set of data. The next step aimed at is the integration of selected algorithms and data preparation tools into CAE-Bench. As soon as this has been accomplished the method needs to be validated on a more substantial data set, i.e. within the working environment of BMW. Then it will be feasible to judge whether the original questions aimed at can be answered.

Acknowledgements

We wish to thank T. Gholami from BMW AG for his support of this work.

References

1. Verbundprojekt AUTO-OPT, funded by the German Ministry for Education and Research, www.auto-opt.de
2. CAE-Bench by MSC.Software, Munich, www.mssoftware.com
3. Kloesgen, W., Zytkow, J., (Eds.): Handbook of Data Mining and Knowledge Discovery. Oxford University Press (2002)
4. Borwein, J., Morales, M.H., Polthier, K., Rodrigues, J.F., (Eds.): Multimedia Tools for Communicating Mathematics. Springer Verlag (2002) ISBN 3-540-42450-4
5. The JavaView Project, www.javaview.de
6. Keim, D.A.: Datenvisualisierung und Data Mining. Datenbank-Spektrum **2** (2002) pp. 30-39
7. Kriegel, H.-P., Brecheisen, S., Januzaj, E., Kröger, P., Pfeifle, M.: Visual Mining of Cluster Hierarchies. Proc. 3rd Int. Workshop on Visual Data Mining, Melbourne, FL, (2003) pp. 151-165.

8. Kriegel, H.-P., Kroger, P., Mashaël, Z., Pfeifle, M., Potke, M., Seidl, Th.: Effective Similarity Search on Voxilised CAD Objects. Proc. 8th Conf. on Database Systems for Advanced Applications, Kyoto, Japan (2003)
9. Devaney, M., Ram, A.: Efficient Feature Selection in Conceptual Clustering. Proc. 14th Int. Conf. on Machine Learning, Morgan Kaufmann Publishers (1997) pp. 92-97
10. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. *Artificial Intelligence*, **97** (1997) pp. 245-271
11. Witten, I.H., Frank, E.: *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco (2000)
12. Weka 3: Data Mining Software in Java, www.cs.waikato.ac.nz/~ml/weka
13. Kuhlmann, A., Thole, C.-A., Trottenberg, U.: AUTOBENCH/AUTO-OPT: Towards an Integrated Construction Environment for Virtual Prototyping in the Automotive Industry. *Lecture Notes in Computer Science* **2840** (2003), pp. 686-690

Pattern Mining Across Domain-Specific Text Collections

Lee Gillam and Khurshid Ahmad

Department of Computing, School of Electronics and Physical Sciences,
University of Surrey, Guildford, GU2 7XH, United Kingdom
{l.gillam, k.ahmad}@surrey.ac.uk

Abstract. This paper discusses a consistency in patterns of language use across domain-specific collections of text. We present a method for the automatic identification of domain-specific keywords – specialist terms – based on comparing language use in scientific domain-specific text collections with language use in texts intended for a more general audience. The method supports automatic production of *collocational networks*, and of networks of *concepts* – thesauri, or so-called ontologies. The method involves a novel combination of existing metrics from work in computational linguistics, which can enable extraction, or *learning*, of these kinds of networks. Creation of ontologies or thesauri is informed by international (ISO) standards in *terminology science*, and the resulting resource can be used to support a variety of work, including data-mining applications.

1 Introduction

A measurable difference appears to exist between language used in specialist communications and language used to communicate with a more general audience. The difference suggests that specialists are relatively disciplined in their language use, and provides the opportunity for automatic processing of natural language texts, for example for identification of the keywords of an *arbitrary* specialisation. On this premise we have explored a method that contrasts frequency lists obtained from predominantly scientific, domain-specific, text collections with lists from general language texts. The method is based on a novel combination of statistical measures originating in corpus linguistics, and is supported by developments in international (ISO) standards in relation to *Terminology Science*, and in the development of ontologies in the computing discipline of *Knowledge Engineering*; such ontologies may be construed as modern-day thesauri.

Approaches to the identification of domain-specific keywords – the *terminology* of the domain – generally rely on extensive prior linguistic knowledge, perhaps embodied in an initially linguistic extraction technique. Our method differs from these approaches in being a primarily statistical treatment that could significantly reduce the amount of prior linguistic knowledge needed. The results of this statistical analysis can be augmented using linguistic techniques such that the entire process bootstraps

itself. The analysis produces collocational networks suitable for use in visualizing sequences of texts [1]. Use of various interchange formats can make such networks available for use as a thesaurus, as a terminology, or as an ontology. In these forms, the results become useful for tasks such as document management or query expansion, and may also provide a means of feature selection for data mining applications involving text collections. The automatic identification of such patterns has long been a goal of terminologists. Approaches to identification of such patterns are generally considered from the perspective of a single collection (corpus) of texts with a single specialism (subject field) in mind. Our work has been undertaken using several such corpora from different specialisms to explore generalisation of the approach.

We consider that since these automatically identified domain-specific keywords – terms – are an artefact of notions, ideas or thoughts – *concepts* – then inter-relationships between the terms provides evidence of the conceptual organisation (ontology or thesaurus) of that domain.

2 Automatically Extracting Terminology / Ontology

While for the Information Retrieval community a *term* seems to be any word that relates to a document or query, stop words aside, for the terminology community, a *term* is “a “verbal designation of a general concept in a specific subject field” (ISO 1087-1, ISO 12620). The phrase “verbal designation” may be misleading, and the phrase “general concept” is the subject of debate, however “specific subject field” indicates the treatment of particular specialisms. In both communities, the notion of statistical significance has been used to identify a term. Statistical significance, for IR purposes, is a function of rarity across a collection of documents: more occurrences of the keyword(s) in fewer documents. Statistical significance for terminology purposes, on the other hand, can be conceived of as a function of rarity in contrast with what is considered to be *general language*. By consideration of this statistical significance, a task variously referred to as *terminology extraction / terminology structuring* [2] or *ontology learning* [3] is possible; an intermediary of this activity may be a *collocational network* [1]. Papers on ontology learning and terminology extraction, and on information extraction, enumerate three techniques: (i) statistical [4], [5]; (ii) linguistic [6]; and (iii) hybrid [7], [8]. Typically, approaches to ontology learning from text employ syntactic parsing [9], [10], [11], [12], [13]. Some authors augment their approach using TF/IDF, word clustering, and coded linguistic relationships [10]. Hybrid techniques may include sophisticated classification systems (neural networks and other learning algorithms), and rely on the frequency counts of linguistic units. Statistical measures, such as log-likelihood and mutual information, may be used to rank information extracted linguistically [8], but predominantly statistical approaches to this task are scarce.

We have developed a statistical method to identify collections of these domain terms – the terminology – automatically, and we use the terms as the basis for thesauri, or the latter-day *ontologies*. These terms are extracted from collections of text in specific subject fields.

2.1 Method

We are interested in the Quirkian notion that frequency of use of words correlates with acceptability of those words as part of the vocabulary [14]:33. Our method uses the 100 million-word British National Corpus (BNC) [15] as reference collection of *general* language. We consider similarities between BNC and specialist corpora as can be derived from Zipf's Law [16], which produces a difference between general language and specialist language and suggests a similarity exists between language use in different specialisms: the approach may be generalisable to other specialist texts. Subsequently, we use a *weirdness* calculation [17], based on one of Malinowski's observations, that has been adapted by *smoothing* [18], to seed a *collocation extraction* technique [19] that results in the production of a network of terms/concepts. By reference to international standards (ISO) for terminology, we can facilitate the construction of terminological resources that have a potentially wider scope of use as thesauri or ontologies. Such a terminology/thesaurus/ontology is then suitable for validation by experts.

For our analysis we considered five specialist text corpora of various sizes: 4 collated at Surrey, consisting of full texts from automotive engineering, nuclear physics, finance and nanoscale science and design, and a fifth from the MuchMore Springer Bilingual Corpus consisting of abstracts from medical journals. Examples presented are generally from one of these corpora – concerned with nanoscale science and design – but similar results have been obtained from all of these corpora, and are the subject of ongoing expert evaluation, and further analysis, in other work. If different specialisms use language in similar ways, it may be possible to systematically extract terminology, or thesauri, or ontology, from *arbitrary* collections of specialist text. In our treatment, we make a distinction between **tokens** (words occurring at various locations in texts) and **types** (the different words used).

2.2 Comparing Text Corpora

The contrast between the general, everyday use of English with that of specialist use has been observed empirically and recently quantified using Zipf's Law. We consider the British National Corpus as an example of general language, alongside five specialist domain corpora, data for which is presented in Table 1.

Table 1. Type and token counts for the 6 corpora that are the subject of our analysis

	Automotive	Nuclear	Finance	Medical	Nanoscale	BNC
Tokens	350920	393993	685037	1081124	1012096	100106029
Types	14252	14937	28793	35033	26861	669417

Counting words in collections of text provides a means by which to study properties of language use. The first 100 most frequently occurring types in a text corpus have been shown to comprise just under half the corpus, which is true of both specialist and non-specialist corpora [20]. The first 100 most frequent types within a specialist text collection of around half a million words comprises between 30 and 40 *open class words (types)* – predominantly nouns specific to the domain. The distinction between open-class words and *closed-class words* – grammatical words, for example

determiners and conjunctions, that tend to appear in stop-lists – requires human judgement and prior (domain) knowledge: a word such as *keep* may occur in a stop-list for IR, but is important in documents relating to *medieval architecture*. Since we seek to automate our approach, we need to determine how language behaves by considering characteristics of language use.

George Kingsley Zipf’s power-law function has been used to show that for a text corpus of N tokens, the rank of a word (type) multiplied by its frequency produces a constant of $N/10$ (e.g. for the Brown Corpus [21]:26-27). Zipf’s law has been tested in analysis of large (sub-)corpora of newswire texts (the Wall Street Journal for 1987, 1988 and 1989: approximately 19 million, 16 million and 6 million tokens respectively) and shows similar behaviour between these collections, with little deviation due to corpus size [22]. In these analyses, Zipf’s law holds only for a certain range of ranks: at high frequency and low frequency similar patterns of deviation from Zipf’s law are encountered that may be accounted for by the Zipf-Mandelbrot law. Our chosen corpora have similar deviations from Zipf’s law, however application of Zipf’s law to words of low frequency - the *hapax legomena* – produced a clear difference between general language and specialist language (Fig.1.).

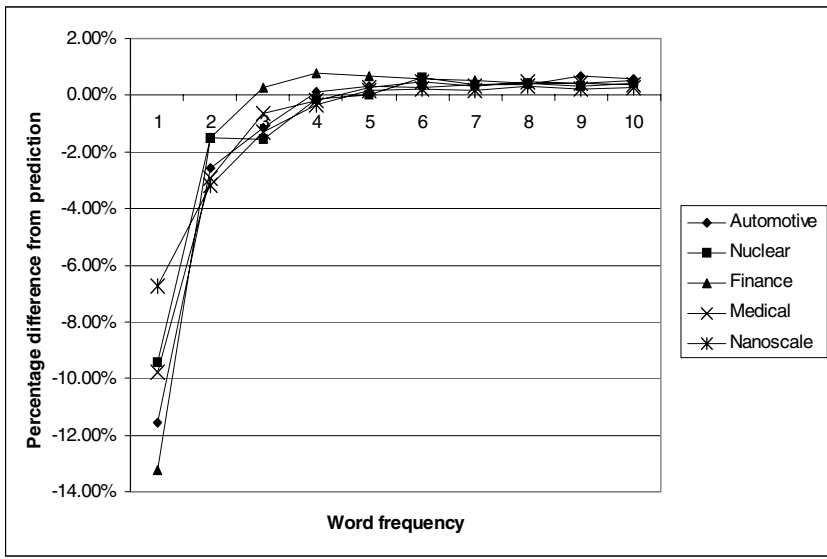


Fig. 1. Difference between percentage of types (y-axis) in the specialist corpora and predicted values derived from Zipf’s law (0.00%) for frequencies between 1 and 10 (x-axis) for 5 specialist corpora. For frequency 1, specialisms have around 37-43% of types compared to expectation (50%) and BNC (53%)

By Zipf’s law, 50% of words should occur at frequency 1. In the BNC, this figure is around 53%, whereas for the specialist corpora only 37-43% of words occur once. Additionally, within this *hapax legomena*, Zipf’s law suggests that 90% of words should occur with frequency of 10 or less. For BNC, this is 84.45%, while it is

around 80% for the specialist corpora. A first inference from such results is that specialist corpora, and the specialists who contribute to them, are disciplined in their language use since they deviate in the same manner from Zipf’s law and from a general language resource such as BNC. A second inference is that the reduced values for specialist corpora fit better with Zipf’s *principle of least effort* [16]: 22-23.

2.3 Automatic Single-Word Term Extraction

The empirical observations above have led us to develop the notion of *weirdness* [17]. On the basis of similar properties of corpora, we compare frequencies of words (types) in the specialist corpora to frequencies in general language. This computation has been adapted using an additive smoothing technique to account for words not found in general language [18]. For the selected specialist corpora, between 13% and 38% of types are not found in the general language contained in the BNC (Table 2).

Table 2. Words (types) in the specialist corpora that do not occur in the reference corpus

Corpus	Types	Tokens	Number of types not in BNC	% types not in BNC
Automotive	14252	350920	1794	13%
Nuclear	14937	393993	4190	28%
Finance	28793	685037	4718	16%
Medical	35033	1081124	11395	33%
Nanoscale	26861	1012096	10231	38%

Weirdness, smoothed, is calculated as

$$weirdness = \frac{N_{GL}f_{SL}}{(1 + f_{GL})N_{SL}} \tag{1}$$

where f_{SL} is the frequency of word in the specialist corpus, f_{GL} is its frequency in BNC, and N_{SL} and N_{GL} are the token counts of the specialist corpus and the BNC respectively. For example, for the Nanoscale corpus, we can produce a list of words and a combination of frequencies and weirdness values which suggest their domain-specificity (Table 3).

Table 3. Words, their frequencies in a specialist corpus and the BNC, and the result of weirdness and its smoothing

Word	Freq	BNC	Weirdness
nanowires	619	0	61225
nanoparticles	829	1	40998
nanowire	360	0	35607
nanotube	969	2	31948
nanoscale	268	0	26508
tunneling	514	1	25420
nanoparticle	232	0	22947

Highly frequent words include closed class words – *the* being most frequent in English; across corpora a weirdness value for *the* of around 1 is obtained. The combination of high frequency and high weirdness is therefore of interest, and can automate removal of these closed class words (stop lists). This may be suitable for stop lists for other languages and purposes also. Resulting lists of frequency and weirdness values can be treated systematically by taking z-scores of each. Where z-score for **both** frequency and weirdness is above a given threshold, the words provided by this mechanism are used in subsequent analysis (Table 4).

Table 4. Number of words selected by z-score thresholds for both frequency and weirdness

	Automotive	Nuclear	Finance	Medical	Nanoscale
z-score					
5	0	0	0	0	1
4	0	0	0	0	5
3	0	1	0	0	6
2	0	3	2	1	8
1	7	6	3	4	19
0	154	176	186	494	352

2.4 Automatic Multiword Term Extraction

While some single-words may be terms in their own right, terms in specialist domains tend to be formed from a number of single words to identify more specific concepts: for example as *multiword terms*. Multiword terms in English generally exclude closed class words, punctuation marks and numerals, although chemical, mathematical and, indeed, nomenclatures for logic use a variety of hyphenation, numerals and other symbols that are important in that subject field. The frequency with which words appear in close proximity to each other has been variously analysed. Magnusson and Vanharanta have created collocational networks for visualising sequences of texts [1] using the “information theoretic concept of mutual information” [23]. Elsewhere, Church’s t-score has been used for calculating strength of association [24]:34. We have found both measures to be limited on three counts: first, the selection of words for treatment by both seems to be arbitrary; second, both metrics take no account of features of the neighbourhood of the selected word(s); third, both metrics consider only two words together. On the first issue, our weirdness-frequency combination seems to offer a solution; for the second and third we consider Smadja’s work on collocations [19]. Smadja uses a neighbourhood of five words and records frequency of words occurring at each position. If the two words consistently appear together in the same relative position in contrast to other positions, this *collocation* is deemed significant. We refer to a process of using such significant collocates as inputs to a subsequent collocation phase as *re-collocation*. This expands a collocation network systematically, depending on the satisfaction of Smadja’s constraints. Table 5 shows a sample of the words that collocate in the five positions either size of *carbon* (frequency of 1506 in about 1 million tokens), in the nanoscale science and design corpus, and that satisfy Smadja’s constraints.

Table 5. Collocations with *carbon* (frequency of 1506) in the Nanoscale science corpus

Collocate	Freq	-5	-4	-3	-2	-1	1	2	3	4	5
nanotubes	690	8	8	9	2	0	647	6	0	7	3
nanotube	252	3	2	2	0	0	229	2	1	5	8
single-walled	77	0	0	1	1	75	0	0	0	0	0
aligned	94	1	1	3	5	74	0	1	1	3	5
multiwalled	70	1	1	2	0	59	0	0	1	5	1
amorphous	58	1	1	6	0	46	0	1	1	0	2
atoms	51	1	2	0	1	0	42	0	1	3	1
nanotips	44	0	2	1	1	0	39	0	0	1	0

Re-collocation of carbon nanotubes produces collocating words such as those in Table 6.

Table 6. Collocations with *carbon nanotubes* (frequency of 647) in the Nanoscale science corpus

Collocate	Frequency	-5	-4	-3	-2	1	1	2	3	4	5
single-walled	73	0	0	1	1	71	0	0	0	0	0
aligned	63	1	1	1	5	48	0	0	2	4	1
multiwalled	53	0	0	1	0	46	0	0	5	1	0
properties	60	1	4	15	32	0	0	0	6	2	0
multiwall	34	0	1	0	1	30	0	2	0	0	0
single-wall	26	0	0	1	0	24	0	0	0	1	0

2.5 Terminology/Ontology Construction

The *collocation network* in our case is a tree branching out from the originally selected list of words. Gillam has shown how this tree can be encoded in conformity

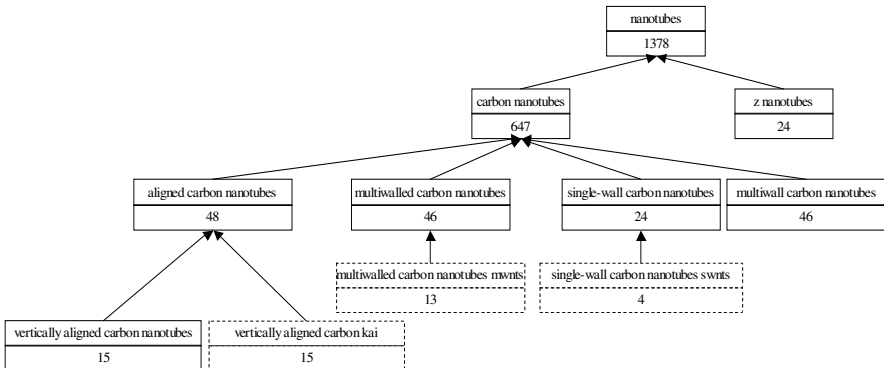


Fig. 2. Fragment of the resulting ontology from a corpus of Nanoscale science and design texts. Dotted outlines denote collocations deemed to be invalid

with international standards for the production of terminology interchange formats using ISO 12620 and ISO 16642 and for use with so-called ontology exchange languages [25]. A fragment of such a tree, resulting from the above analysis, and suitable for such encoding, is shown in Fig. 2.

3 Discussion

The method discussed relates to Minsky's "thesaurus problem" [26]:27 of building and maintaining a thesaurus useful for a specific task or set of tasks, of learning to build thesauri, and of finding "new ways [to] make machines first to use them, then to modify them, and eventually to build for themselves new and better ones". A thesaurus has been described as "one type of ontology, one specialized to information retrieval" [27], and is built to show the inter-relationships between words. Unlike a dictionary, which is usually alphabetically organized and seeks to elaborate meaning, a thesaurus is developed to elaborate the conceptual (hierarchical and part-whole) relationships between a word, or more specifically a word related to a concept, and one or more other words. Our method for constructing ontologies extends work on construction of *collocational networks*. These ontologies, or terminologies, or thesauri, can be extracted automatically from text collections, and show how knowledge in a specific subject field is organised. Such resources may be useful for the organization of this scientific information. Other applications include: generating buy/sell signals in financial trading [28], health policy communication [29], digital heritage [30] and query expansion for multimedia information retrieval [31], [32]. The resulting ontology has been exported to a *de facto* standard ontology editor – Protégé – for viewing and editing. Here it becomes the basis for developing intelligent (rule-based systems) using applications such as JESS [33]. The ontology also enables text-based *feature selection* to be made, which may be useful for systems such as WebSOM [34].

Initial discussions with domain experts have validated the first results with some degree of confidence, and we are studying the effects of increasing the length of multiword patterns being generated, against decrease in frequency. For example, at low frequencies that are not statistically validated, we have *aligned single-walled carbon nanotubes* (2) and *large-diameter single-walled carbon nanotubes* (2). Subsequent effort is required to classify collocations into *facets*: in the examples presented, the types of carbon nanotubes appear to have "walledness" as a significant facet, and being *aligned* has importance also, though is perhaps a value of a different facet. Determining this distinction currently requires expert input. *properties* is not a positionally valid collocation – though we can infer that *properties of carbon nanotubes* are described in this collection. We have considered combined use with linguistic patterns elsewhere [35].

Acknowledgements. This work was supported in part by research projects sponsored by the EU (SALT: IST-1999-10951, GIDA: IST-2000-31123, LIRICS: eContent-22236) and by UK research councils: EPSRC (SOCIS: GR/M89041/01, REVEAL: GR/S98450/01) and ESRC (FINGRID: RES-149-25-0028).

References

1. Magnusson, C. and Vanharanta, H.: Visualizing Sequences of Texts Using Collocational Networks. In Perner, P. and Rosenfeld, A. (Eds): *MLDM 2003*, LNAI 2734 Springer-Verlag, Heidelberg. (2003) 276-283
2. Grabar, N. and Zweigenbaum, P.: Lexically-based terminology structuring. *Terminology* 10(1). John Benjamins, Amsterdam (2004) 23-53.
3. Maedche, A.: *Ontology Learning for the Semantic Web*. The Kluwer International Series in Engineering and Computer Science, Vol. 665. Kluwer Academic Publishers (2002).
4. Salton, G.: Experiments in Automatic Thesauri Construction for Information Retrieval. In *Proceedings of the IFIP Congress*, Ljubljana, Yugoslavia. Vol. TA-2. (1971) 43-49.
5. Jing, Y. and Croft, W.B.: An Association Thesaurus for Information Retrieval. In Bretano, F., Seitz, F. (eds.), *Proc. of RIAO'94 Conference*, CIS-CASSIS, Paris, France (1994) 146-160.
6. Grefenstette, G.: *Explorations in Automatic Thesaurus Discovery*. Boston, USA: Kluwer Academic Publishers (1994)
7. Drouin, P.: Term extraction using non-technical corpora as a point of leverage. *Terminology* 9(1). John Benjamins, Amsterdam (2003) 99-115.
8. Vivaldi, J. and Rodríguez, H.: Improving term extraction by combining different techniques. *Terminology* 7(1). John Benjamins, Amsterdam (2001) 31-47.
9. Maedche, A. and Volz, R.: *The Ontology Extraction and Maintenance Framework Text-To-Onto*. Workshop on Integrating Data Mining and Knowledge Management. California, USA (2001)
10. Maedche, A. and Staab, S.: *Ontology Learning*. In S. Staab & R. Studer (eds.): *Handbook on Ontologies in Information Systems*. Heidelberg: Springer (2003).
11. Faure, D. and Nédellec, C.: Knowledge Acquisition of Predicate Argument Structures from Technical Texts Using Machine Learning: The System ASIUM. LNCS 1621. Springer-Verlag, Heidelberg. (1999) 329-334.
12. Faure, D. and Nédellec, C.: ASIUM: Learning subcategorization frames and restrictions of selection. In Y. Kodratoff, (Ed.), *10th Conference on Machine Learning (ECML 98)*, Workshop on Text Mining, Chemnitz, Germany. (1998).
13. Mikheev, A. and Finch, S.: A Workbench for Acquisition of Ontological Knowledge from Natural Text. In *Proc. of the 7th conference of the European Chapter for Computational Linguistics (EACL'95)*, Dublin, Ireland. (1995) 194-201.
14. Quirk, R.: *Grammatical and Lexical Variance in English*. Longman, London & New York (1995)
15. Aston, G. and Burnard, L.: *The BNC Handbook: Exploring the British National Corpus*. Edinburgh University Press (1998).
16. Zipf, G.K.: *Human Behavior and the Principle of Least Effort*. Hafner, New York. (1949).
17. Ahmad, K. and Davies, A.E.: Weiridness in Special-language Text: Welsh Radioactive Chemicals Texts as an Exemplar. *Internationales Institut für Terminologieforschung Journal* 5(2). (1994) 22-52.
18. Gale, W. and Church, K. W.: What's wrong with adding one? In Oostdijk, N. and de Haan, P. (eds.): *Corpus-Based Research into Language: In honour of Jan Aarts*. Rodopi, Amsterdam (1994), 189-200
19. Smadja, F.: Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1). Oxford University Press. (1993), 143-178

20. Ahmad, K.: Neologisms to Describe Neologisms: Philosophers of Science and Terminological Innovation. In (Ed.) Peter Sandrini: Proc. of Terminology and Knowledge Engineering (1999), 54-73.
21. Manning, C. and Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA. (1999)
22. Ha, L. Q., Sicilia, E. , Ming, J. and Smith, F. J.: Extension of Zipf's law to words and phrases. In Proceedings of International Conference on Computational Linguistics (COLING 2002), Taipei, Taiwan. (2002), 315-320
23. Church, K.W. and Hanks, P.: Word association norms, mutual information and lexicography. In Proceedings of the 27th Annual Conference of the Association of Computational Linguistics (1989), 76-82.
24. Jacquemin, C.: Spotting and Discovering Terms through Natural Language Processing. MIT Press. Cambridge, MA. (2001)
25. Gillam, L.: Systems of concepts and their extraction from text. Unpublished PhD thesis, University of Surrey. (2004).
26. Minsky, M.: Semantic Information Processing. MIT Press (1968)
27. Oard, D.W.: Alternative approaches for cross-language text retrieval. In AAAI Symposium on Cross-Language Text and Speech Retrieval. American Association for Artificial Intelligence. (1997).
28. Gillam, L. (Ed): Terminology and Knowledge Engineering: making money in the financial services industry. Proceedings of workshop at 2002 conference on Terminology and Knowledge Engineering (2002).
29. Gillam, L. and Ahmad, K.: Sharing the knowledge of experts. *Fachsprache* 24(1-2). (2003), 2-19.
30. Gillam, L., Ahmad, K. Salway,,: Digital Heritage and the use of Terminology. Proceedings of Terminology and Knowledge Engineering. (2002)
31. Ahmad, K., Tariq, M., Vrusias, B. and Handy, C.: Corpus-Based Thesaurus Construction for Image Retrieval in Specialist Domains. ECIR 2003, LNCS 2633. Springer Verlag, Heidelberg (2003), 502-510.
32. Vrusias, B. Tariq, M. and Gillam, L.: Scene of Crime Information System: Playing at St Andrews. CLEF 2003, LNCS 3273. Springer Verlag, Heidelberg (2004), 631-645.
33. Eriksson, H.: Using JessTab to Integrate Protégé and Jess. *IEEE Intelligent Systems* 18(2). (2003), 43-50
34. Kohonen, T., Kaski, S., Lagus, K. Salojärvi, J., Honkela, J., Paatero, V. and Saarela, A.: Self Organization of a Massive Document Collection. *IEEE Transactions on Neural Networks* 11(3). (2000), 574-585.
35. Gillam, L., Tariq, M. and Ahmad, K.: Terminology and the Construction of Ontology. *Terminology*. John Benjamins, Amsterdam. *Terminology* 11:1 (2005), 55-81.

Text Classification Using Small Number of Features

Masoud Makrehchi and Mohamed S. Kamel

Pattern Analysis and Machine Intelligence Lab,
Department of Electrical and Computer Engineering,
University of Waterloo, Waterloo, Ontario N2L 3G1, Canada
{makrechi, mkamel}@pami.uwaterloo.ca

Abstract. Feature selection method for text classification based on information gain ranking, improved by removing redundant terms using mutual information measure and inclusion index, is proposed. We report an experiment to study the impact of term redundancy on the performance of text classifier. The result shows that term redundancy behaves very similar to noise and may degrade the classifier performance. The proposed method is tested on an SVM text classifier. Feature reduction by this method remarkably outperforms information gain based feature selection.

1 Introduction

Recently text classification has been one of the fast paced applications of machine learning and data mining [1]. There are many applications using text classification techniques such as natural language processing and information retrieval [2]. Since text classification is a supervised learning process, a wide range of learning methods, namely nearest neighbour, regression models, Bayesian approach, decision trees, inductive rule learning, neural networks and support vector machines have been proposed [3, 4].

Most text classification algorithms use vector space model or bag of words to represent text documents. In this model, every word or group of words, depends on working with a single word or a phrase, called a term, which represents one dimension of the feature space. A positive number is assigned to each term. This number can be the frequency of the term in the text [5].

One problem with this modelling is high dimensionality of feature space, meaning a very large vocabulary that consists of all terms occurring at least once in the collection of documents. Although high dimensional feature space has destructive influences on the performance of most text classifiers, its impact on increasing complexity is worse and expensive. Then, two main objectives of feature selection are improving both classification effectiveness and computational efficiency. [6, 7].

In aggressive feature selection, most irrelevant, non-predictive, and non-informative features are removed and classification task is performed by very

few features with minimum loss of performance and maximum reduction of complexity. In [6] the number of selected features is as low as 3% of features. More aggressive feature selection, including only 1% of all features, has been reported in [8]. Both reports are about feature selection for text classifiers. In this type of feature selection strategies, the main concern is the complexity reduction, as well as improving the classifier performance.

One well-known approach for removing a large number of non-predictive features is feature ranking [6, 8]. Being ranked by a scoring metric such as information gain, Chi-Squared or odds-ratio, all features are descendingly sorted and a very few number of best features are kept and the rest of features are removed. However, these methods have a serious disadvantage, which is ignoring the correlation between terms because most ranking measures consider the terms individually. An experiment, detailed in the next section, shows that the impact of term redundancy is as distractive as noise.

In this paper, a new approach for feature selection, with more than 98% reduction, is proposed. The method is based on a multi stage feature selection including pre-processing tasks, information gain based term ranking and removing redundant terms by a proposed method which uses mutual information measure and inclusion index. The paper consists of five sections. After the introduction, impact of redundancy on the performance of text classifier is discussed in Section 2. In Section 3, the proposed multi stage feature reduction and a method to identify and remove redundant terms are introduced. Experimental results and conclusion are presented in Sections 4 and 5, respectively.

2 Impact of Redundancy on the Performance of Text Classifiers

Redundancy is a kind of data dependency and correlation which can be estimated by different ways, such as the Jaccard distance, Cosin similarity, co-occurrence and co-location measures [9, 10, 11]. In this paper, redundancy between two terms is measured by mutual information. An experiment is set up in order to illustrate the influence of redundancy on the classifier performance. An SVM classifier with a linear kernel is employed. The data collection is the well known 20 Newsgroups data set. In this experiment, classification accuracy is used as a performance evaluation measure. We show that adding redundancy, in the case of very low number of features, can degrade the accuracy. The testing process is as follows.

Let $\mathbf{T} = \{t_1, t_2, \dots, t_N\}$ be the vocabulary. The terms are ranked by information gain, such that t_1 is the best term and t_N is the worst one. A smaller set \mathbf{V} , so called the set of selected features, is defined as follows; $\mathbf{V} = \{v_1, v_2, \dots, v_n\}$, $\mathbf{V} \subset \mathbf{T}$, $n \ll N$. Three different forms of V are generated by the following schemas;

- n best terms: The n first terms of \mathbf{T} are selected such that $v_i = t_i, 1 \leq i \leq n$.
- $n/2$ best terms + $n/2$ redundant terms: In this schema, vector \mathbf{V} contains two parts. First part is selected like first schema, except instead of n , $n/2$ best terms are picked up. The $n/2$ terms in the second part are artificially

Table 1. The impact of redundancy and noise on the accuracy of the SVM text classifier

number of terms	5	10	15	20	25	30	35	40	average
<i>n</i> best terms	0.1793	0.3465	0.5843	0.6991	0.7630	0.8455	0.9299	0.9369	0.6606
50% redundancy	0.1493	0.2499	0.3473	0.4456	0.5029	0.5922	0.6646	0.6925	0.4555
50% noise	0.1485	0.2483	0.3302	0.4038	0.5024	0.5833	0.6752	0.7185	0.4513

generated by adding very small noise to each term of the first part. By this formulation, the rate of redundancy is at least 50%. Since we use binary features (without weights), added noise is a uniform binary noise changing the corresponding binary features from zero to one or vice versa. In order to achieve high degree of redundancy, few number of features, 2% of whole features, are chosen to be affected by noise.

- $n/2$ best terms + $n/2$ noise: It is the same as previous schema except the second part consists of noisy terms. Because of using feature ranking measures, $n/2$ last (worst) terms which can be treated as noise, are added to the first part.

All three feature vectors with different values for n , $n = \{5, 10, \dots, 40\}$, are submitted to the SVM classifier. In order to estimate the accuracy, a five-fold cross validation schema is employed. In this process, the collection is divided into five subsets. The experiment is repeated five times. Each time we train the classifier with four subsets and leave the fifth one for test phase. The average of five accuracies is the estimated accuracy.

Table 1 illustrates the result. It clearly shows that redundancy and noise reduce the accuracy. Comparing the averages depicts both schemas have almost similar impact on the classifier. In a small ranked feature vector, the risk of having redundant term is quite high. For example in a five-term feature vector, if there is only one redundant term, we are actually using four terms instead of five because one of the terms is useless. By removing the redundant term, we make room for another term which can improve the predictive power of the feature vector.

3 Proposed Approach

The main goal of the proposed schema is providing a solution for feature selection with a high rate of reduction, by which the number of selected features \mathbf{V} is much less than those in the original vocabulary \mathbf{T} . We propose a three-stage feature selection strategy including pre-processing tasks, information gain ranking, and removing redundant terms. The first stage involves pre-processing tasks that include Porter word stemming which can reduce almost 40% of terms, removing general stopwords reducing about 200 terms, and removing most and least frequent terms. Since we are using the 20 Newsgroups data set, the original vocabulary has about 118,275 terms. The pre-processing tasks cut down the size of the vocabulary 75.50%. In this step, we are not losing much information

because the pre-processing tasks remove non-informative, noise, stopwords, and misspelled words.

In the second stage, information gain is used to select most informative and predictive terms. Information gain is one of the most efficient measures for feature ranking in classification problems [8]. Yang and Pedersen [7] have shown that sophisticated techniques such as information gain or Chi-Squared can reduce the dimensionality of the vocabulary by a factor of 100 with no loss (or even with a small increase) of effectiveness. Here, the terms in the vocabulary after pre-processing which includes 28,983 terms, are ranked by information gain. The 10% of best terms are chosen as most informative and predictive terms.

Information gain and other filter based feature selection methods ignore the correlation between features and evaluate them individually. The main motivation of the work reported in this paper is improving information gain ranking by identifying any correlation between terms, and extracting and removing redundancies, which is the third stage. At this level, about 5% to 20% of ranked features are selected. While employing very few features, any term redundancy influences the output of the classifier and reduces the accuracy. The proposed approach has two core elements; mutual information and inclusion index which are detailed in the following subsections.

3.1 Mutual Information

Mutual information is a measure of statistical information shared between two probability distributions. Based on the definition in [12], mutual information $I(x; y)$ is computed by the relative entropy of a joint probability distribution like $p(x, y)$ and the product of the marginal probability distributions $p(x)$ and $p(y)$

$$I(x; y) = D(p(x, y) || p(x)p(y)) = \sum_x \sum_y p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \quad (1)$$

Mutual information has been applied in text mining and information retrieval for applications like word association [13] and feature selection [14]. Mutual information is viewed as the entropy of co-occurrence of two terms when observing a category. We practically compute mutual information between two other mutual information measures. Each measure represents shared information between a term like t_i and a class such as c_k . Since we are interested in the distribution of a pair of terms given a specific category, the joint distribution is considered as the probability of occurrence of the two terms t_i and t_j in those documents belonging to the class c_k . Eq. 1 can be rewritten as follows

$$I(t_i; c_k) = \sum_{t_i} \sum_{c_k} p(t_i, c_k) \log \frac{p(t_i, c_k)}{p(t_i)p(c_k)} \quad (2)$$

where $I(t_i; c_k)$ is the mutual information of the distribution of term t_i and category c_k . Mutual information itself has been used as a ranking measure and showed very poor result [7]. Eq. 2 might be written for term t_j exactly the same way. In other word, $I(t_i; c_k)$ is the entropy of $p(t_i \cap c_k)$ which is the probability

distribution of the occurrence of the term t_i in the class c_k . The total mutual information (φ) is calculated as follows

$$\varphi \{I(t_i; c_k); I(t_j; c_k)\} = \varphi(t_i \cap c_k, t_j \cap c_k) \tag{3}$$

$$\varphi(t_i \cap c_k, t_j \cap c_k) = \sum_{t_i, c_k} \sum_{t_j, c_k} p(t_i \cap c_k, t_j \cap c_k) \log \frac{p(t_i \cap c_k, t_j \cap c_k)}{p(t_i \cap c_k) \cdot p(t_j \cap c_k)} \tag{4}$$

$\varphi \{I(t_i; c_k); I(t_j; c_k)\}$ is a point-wise mutual information. The total mutual information of two terms when observing whole category information is the average of the mutual information over c . This measure is simply represented by the summarized form $\varphi(t_i; t_j)$.

$$\varphi(t_i; t_j) = \sum_{k=1}^C \varphi(t_i \cap c_k, t_j \cap c_k) \tag{5}$$

where C is the number of categories. Since φ has no upper bound, normalized mutual information Φ which has upper bound and is a good measure to compare two shared information, is proposed as follows [16].

$$\Phi(t_i; t_j) = \frac{\varphi(t_i; t_j)}{\sqrt{I(t_i; c) \cdot I(t_j; c)}}, 0 \leq \Phi(t_i; t_j) \leq 1 \tag{6}$$

From [16], φ and $I(t_i; c)$ can be estimated as following equations,

$$I(t_i; c) = \sum_{k=1}^C \frac{n_{t_i}^{c_k}}{n} \log \frac{\frac{n_{t_i}^{c_k}}{n}}{\frac{n_{t_i}}{n} \cdot \frac{n_{c_k}}{n}} = \frac{1}{n} \sum_{k=1}^C n_{t_i}^{c_k} \log \frac{n \cdot n_{t_i}^{c_k}}{n_{t_i} \cdot n_{c_k}} \tag{7}$$

$$\varphi(t_i; t_j) = \sum_{k=1}^C \frac{n_{t_i, t_j}^{c_k}}{n} \log \frac{\frac{n_{t_i, t_j}^{c_k}}{n}}{\frac{n_{t_i, t_j}}{n} \cdot \frac{n_{c_k}}{n}} = \frac{1}{n} \sum_{k=1}^C n_{t_i, t_j}^{c_k} \log \frac{n \cdot n_{t_i, t_j}^{c_k}}{n_{t_i, t_j} \cdot n_{c_k}} \tag{8}$$

where n is the total number of documents in the collection, n_{c_k} depicts the number of documents in k^{th} category, n_{t_i} (n_{t_i, t_j}) is the number of documents which have term t_i (both t_i and t_j). The number of documents which belongs to the k^{th} class, and includes the term t_i (t_i and t_j) is represented by $n_{t_i}^{c_k}$ ($n_{t_i, t_j}^{c_k}$). Eq. 6 is estimated as follows,

$$\Phi(t_i; t_j) = \frac{\sum_{k=1}^C n_{t_i, t_j}^{c_k} \log \frac{n \cdot n_{t_i, t_j}^{c_k}}{n_{t_i, t_j} \cdot n_{c_k}}}{\sqrt{\sum_{k=1}^C n_{t_i}^{c_k} \log \frac{n \cdot n_{t_i}^{c_k}}{n_{t_i} \cdot n_{c_k}} \cdot \sum_{k=1}^C n_{t_j}^{c_k} \log \frac{n \cdot n_{t_j}^{c_k}}{n_{t_j} \cdot n_{c_k}}}} \tag{9}$$

Φ is equal to one if the two terms are completely identical and correlated when observing a category, and $\Phi = 0$ if the two terms are completely uncorrelated. It should be noted that although point-wise mutual information $\varphi \{I(t_i; c_k); I(t_j; c_k)\}$ can be negative [15], the average mutual information

$\varphi(t_i; t_j)$ is always positive and its normalized version is less than or equal to one.

Φ is calculated for all possible pairs of terms in the vocabulary. The result is Φ matrix in the order of $M \times M$, where M is the size of the vocabulary or the number of terms. Since Φ is a symmetric measure, and always $\Phi(t_i; t_i) = 1$, in order to construct the matrix, $\frac{M(M-1)}{2}$ number of Φ calculations are necessary, that it very expensive. One approach to overcome the problem is to calculate Φ matrix for a very small subset of terms S of the vocabulary V . It means instead of the full Φ matrix, a sub-matrix of Φ is provided. In other words, we need to calculate Φ for most likely correlated terms. Let us suppose that there are n_s groups of correlated terms in the vocabulary. The problem is identifying these groups and calculating Φ for each of them. We propose inclusion index and matrix for this purpose.

3.2 Inclusion Index

Let $D = \{d_1, d_2, \dots, d_n\}$ be the collection of documents. Every document is represented by a vector of words, called the document vector space, for example,

$$d_k = \{w_{k,1}.t_1, w_{k,2}.t_2, \dots, w_{k,M}.t_M\} \quad (10)$$

where $w_{k,q}$ is the weight of the q^{th} term in the k^{th} document. Here we use binary weighting which depends on whether the term is in the document or not. As a consequence, D can be represented by an $N \times M$ matrix in which every row (d_k) is a document and every column (t_i) represents the occurrence of the term in every document. Based on this notation, inclusion, which is a term-term relation, is defined in [17]. Inclusion index $Inc(t_i, t_j)$, representing how much t_j includes t_i , is calculated by,

$$Inc(t_i, t_j) = \frac{||t_i \cap t_j||}{||t_i||}, \quad Inc(t_i, t_j) \neq Inc(t_j, t_i) \quad (11)$$

where $||\cdot||$ is the cardinal number of the set. $Inc(t_i, t_j) = 1$ when t_j is completely covering t_i and called full inclusive. $Inc(t_i, t_j) = 0$ means there is no overlap between the two terms. There is also partial inclusion when $0 < Inc(t_i, t_j) < 1$. t_j is called more inclusive than t_i if $Inc(t_i, t_j) > Inc(t_j, t_i)$. The inclusion matrix **Inc** is an $M \times M$ matrix in which each entry is an inclusion index between two terms.

3.3 Redundancy Removal Algorithm

The main idea in identifying redundant terms is finding the sets of correlated terms. For example, {rec,hockey,motorcycl,bike,nhl,playoff} shows one of these sets including six correlated terms. The sets are extracted using inclusion matrix **Inc**. Algorithm 1 represents the detail of extracting the sets and then identifying redundant terms.

Let S_q be the q^{th} set of correlated terms. Instead of calculating full matrix of Φ , it is only obtained for the terms in the S_q . The resulting matrix is represented

Algorithm 1 Extracting redundant terms

```

for  $1 \leq i, j \leq M$  if  $Inc(i, j) > threshold \Rightarrow inc(i, j) \leftarrow 1$  else  $inc(i, j) \leftarrow 0$ 
for  $1 \leq i \leq M$   $TermIndex(i) \leftarrow 0$ 
 $i \leftarrow 1$ 
while “the set of zero element in  $TermIndex$  is not empty”
     $k \leftarrow$  index of 1st zero element in  $TermIndex$ 
     $TermIndex(k) \leftarrow 1, u \leftarrow k, l \leftarrow 1$ 
    while “ $l$  is non-zero”
         $z \leftarrow$  the set of non-zero elements of  $k^{th}$  column of  $\mathbf{inc}$ 
        if “ $z$  is non-empty”  $\Rightarrow$  append  $z$  to  $u$ , sort  $u$ 
         $TermIndex(k) \leftarrow 1$ 
         $x \leftarrow$  number of zero elements of  $TermIndex$  according to  $u$ 
         $l \leftarrow$  number of elements in  $x$ 
        if  $l > 0 \Rightarrow k \leftarrow u(x(1))$ 
    end while
     $CorrelatedTermSet(i) \leftarrow u$ 
     $i \leftarrow i + 1$ 
end while
remove all sets from  $CorrelatedTermSet$  which have less than two elements
for  $q = 1$  to number of set of correlated terms
    calculate  $\Phi_q$ , calculate  $\mathbf{Inc}_q$ 
    for  $i = 1$  to number of elements in  $q^{th}$  set of correlated terms
        for  $j = 1$  to number of elements in  $q^{th}$  set of correlated terms
             $R_q(i, j) \leftarrow Inc_q(i, j) \cdot \Phi_q(i, j)$ 
            if  $i = j \Rightarrow R_q(i, j) \leftarrow 0$ 
        end for
    end for
    keep maximum element of each row of  $\mathbf{R}_q$  and make other else zero
     $RedundantTerms \leftarrow$  terms according to the whole zero columns of  $\mathbf{R}_q$ 
end for

```

by Φ_q . We do the same for \mathbf{Inc}_q . Matrix \mathbf{R}_q , which is called redundancy matrix, is calculated by entry-entry multiplication of Φ_q and \mathbf{Inc}_q as follows

$$R_q(i, j) = \Phi_q(i, j) \cdot Inc_q(i, j), \quad 1 \leq i, j \leq n_q \tag{12}$$

where n_q is the number of terms in S_q . The i^{th} row of \mathbf{R}_q , which is an $n_q \times n_q$ matrix, shows that the i^{th} term (in S_q) in which terms is included or with which ones are being covered. In each row the maximum entry is kept and the others are set to zero. Finally, every term that its corresponding column in \mathbf{R}_q is full zero (all elements are zero), is assigned as a redundant term because it does not include any other term. Table 2 shows the resulting matrices for a set of correlated terms.

4 Experimental Results

The proposed approach has been applied on 20 Newsgroups data set using an SVM classifier with linear kernel. Although there are some reports showing fea-

Table 2. An example of extracting redundant terms from q^{th} set of correlated terms, (A) normalized mutual information matrix Φ_q , (B) inclusion sub-matrix \mathbf{Inc}_q , (C) multiplication of the two matrices (Φ_q and \mathbf{Inc}_q), (D) term redundancy matrix \mathbf{R}_q . Based on \mathbf{R}_q , all terms, whose corresponding columns are zero, are redundant and should be removed

(A)						
	rec	hockey	motorcycl	bike	nhl	playoff
rec	1	0.4448	0.4415	0.2866	0.2078	0.2059
hockey	0.4448	1	0	0	0.4555	0.4300
motorcycl	0.4415	0	1	0.5886	0	0
bike	0.2866	0	0.5886	1	0	0
nhl	0.2078	0.4555	0	0	1	0.1754
playoff	0.2059	0.4300	0	0	0.1754	1

(B)						
	rec	hockey	motorcycl	bike	nhl	playoff
rec	1	0.2221	0.2255	0.1162	0.0669	0.0680
hockey	0.9951	1	0	0	0.2998	0.2883
motorcycl	0.9903	0	1	0.4911	0	0
bike	0.9906	0	0.9530	1	0	0
nhl	0.9945	0.9945	0	0	1	0.2623
playoff	1	0.9459	0	0	0.2595	1

(C)						
	rec	hockey	motorcycl	bike	nhl	playoff
rec	0	0.0988	0.0995	0.0333	0.0139	0.0140
hockey	0.4426	0	0	0	0.1366	0.1240
motorcycl	0.4372	0	0	0.2891	0	0
bike	0.2839	0	0.5609	0	0	0
nhl	0.2067	0.4530	0	0	0	0.0460
playoff	0.2059	0.4067	0	0	0.0455	0

(D)						
	rec	hockey	motorcycl	bike	nhl	playoff
rec	0	0	0.0995	0	0	0
hockey	0.4426	0	0	0	0	0
motorcycl	0.4372	0	0	0	0	0
bike	0	0	0.5609	0	0	0
nhl	0	0.4530	0	0	0	0
playoff	0	0.4067	0	0	0	0

ture selection for SVM classifier not only is unnecessary but also can reduce its performance [6,18], in addition to [8], in this paper we show that for a very small size of feature vector, SVM performance can be improved by feature selection through redundancy reduction.

The proposed schema has been evaluated by comparing its results with those of stand-alone information gain ranking. A five-fold cross validation is used for better estimation of classifier performance. In addition to classifier accuracy, two more performance indices have been used, including micro-average, and macro-average. They are calculated based on a_j , the number of samples which are correctly classified as class j , and b_j , the number of samples wrongly classified as class j .

$$macro - average = \frac{\sum_{i=1}^C \frac{a_i}{a_i + b_i}}{C}, \quad micro - average = \frac{\sum_{i=1}^C a_i}{(\sum_{i=1}^C a_i) + (\sum_{i=1}^C b_i)} \quad (13)$$

where C is the number of categories. Table 3 presents the results of two methods with different performance measures. Each method has been applied on the SVM

Table 3. Comparing the results of aggressive feature selection using information gain ranking and the proposed method (bold) for the SVM text classifier

number of terms	5	10	15	20	25	30	35	40	average
accuracy	0.1793	0.3465	0.5843	0.6991	0.7630	0.8455	0.9299	0.9369	0.6606
	0.2028	0.4854	0.7098	0.8032	0.9031	0.9036	0.9027	0.9022	0.7266
micro-average	0.1113	0.3334	0.5567	0.6799	0.7391	0.8481	0.9331	0.9403	0.6427
	0.1601	0.4645	0.6860	0.7690	0.8844	0.8988	0.8871	0.8845	0.7043
macro-average	0.1004	0.2932	0.5065	0.6470	0.7185	0.8196	0.9300	0.9370	0.6190
	0.1260	0.4120	0.6610	0.7640	0.8826	0.8842	0.8822	0.8827	0.6868

classifier with eight levels of aggressive feature selections. In all measures, and most feature selection levels, the proposed method has outperformed information gain ranking. The last column of the table depicts the averages which clearly show that the proposed approach is more efficient.

5 Conclusion

Aggressive feature selection, with higher than 95% feature reduction, was discussed. This sort of feature selections is very applicable to text classifiers while because of dealing with huge size of feature space so called vocabulary. Text classifiers, working with very small feature vectors, are very sensitive to noise, outliers and redundancies. Then, improving any classical feature selection method like feature ranking for aggressive reduction is strongly necessary.

Term redundancy in text classifiers causes a serious drawback in most feature rankings, such as information gain, because they always ignore correlation between terms. The result of an experiment in the paper showed that the effect of term redundancy can be worse than noise. To find and reduce term redundancy, a method was proposed for improving aggressive feature selection by information gain ranking. The method was based on identifying and removing term redundancy using mutual information measure and inclusion index. Terms were grouped in a few sets of correlated terms using inclusion matrix. In the next step each set was modelled by the term redundancy matrix.

Aggressive feature selection approaches by stand-alone information gain ranking and proposed method (removing the redundant term from ranked feature vector by information gain) were compared in an SVM text classifier framework. Results showed that with three evaluation measures, the proposed schema outperformed the aggressive feature selection by the stand-alone information gain. The proposed method improved information gain 10% in accuracy, 9.5% in macro-average, 11% in micro-average. Better results are expected for other feature ranking methods such as Chi-Squared and odds-ratio.

Acknowledgement

This research was supported in part by the National Science and Engineering Research Council of Canada.

References

1. Sebastiani, F.: Machine learning in automated text categorization. *ACM Computing Surveys* **34** (2002) 1–47
2. Lam, W., Ruiz, M.E., Srinivasan, P.: Automatic text categorization and its applications to text retrieval. *IEEE Transactions on Knowledge and Data Engineering* **11** (1999) 865–879
3. Berry, M.W.: *Survey of Text Mining: Clustering, Classification, and Retrieval*. Springer (2004)
4. Yiming, Liu, X.: A re-examination of text categorization methods. In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. (1999) 42–49
5. Wong, S.K.M., Raghavan, V.V.: Vector space model of information retrieval: a reevaluation. In: *Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval*. (1984) 167–185
6. Rogati, M., Yang, Y.: High-performing feature selection for text classification. In: *Proceedings of the eleventh international conference on Information and knowledge management*. (2002) 659–661
7. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In Fisher, D.H., ed.: *Proceedings of ICML-97, 14th International Conference on Machine Learning*, Nashville, US (1997) 412–420
8. Gabrilovich, E., Markovitch, S.: Text categorization with many redundant features: Using aggressive feature selection to make svms competitive with c4.5. In: *Proceedings of ICML-04, Twenty-first international conference on Machine learning*, Banff, Alberta, Canada (2004) 321–328
9. Dagan, I., Lee, L., Pereira, F.C.N.: Similarity-based models of word cooccurrence probabilities. *Machine Learning* **34** (1999) 43–69
10. Xu, J., Croft, W.B.: Corpus-based stemming using cooccurrence of word variants. *ACM Trans. Inf. Syst.* **16** (1998) 61–81
11. Soucy, P., Mineau, G.W.: A simple feature selection method for text classification. In Nebel, B., ed.: *Proceeding of IJCAI-01, 17th International Joint Conference on Artificial Intelligence*, Seattle, US (2001) 897–902
12. Cover, T.M., Thomas, J.A.: *Elements of information theory*. Wiley-Interscience (1991)
13. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. *Comput. Linguist.* **16** (1990) 22–29
14. Wang, G., Lochovsky, F.H.: Feature selection with conditional mutual information maximin in text categorization. In: *CIKM '04: Proceedings of the Thirteenth ACM conference on Information and knowledge management*. (2004) 342–349
15. Mackay, D.: *Information Theory, Inference and Learning Algorithms*. Cambridge University (2003)
16. Strehl, A., Ghosh, J.: Cluster ensembles – a knowledge reuse framework for combining partitionings. In: *Proceedings of AAAI 2002, Edmonton, Canada, AAAI (2002)* 93–98
17. Salton, G.: Recent trends in automatic information retrieval. In: *SIGIR '86: Proceedings of the 9th annual international ACM SIGIR conference on Research and development in information retrieval*. (1986) 1–10
18. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In Nédellec, C., Rouveirol, C., eds.: *Proceedings of ECML-98, 10th European Conference on Machine Learning*. Number 1398, Chemnitz, DE, Springer Verlag, Heidelberg, DE (1998) 137–142

Low-Level Cursive Word Representation Based on Geometric Decomposition

Jian-xiong Dong¹, Adam Krzyżak³, Ching Y. Suen², and Dominique Ponson¹

¹ IMDS Software, 75 rue Queen, Suite 6200, Montréal, Québec, H3C 2N6
{jxdong, dponson}@imds-world.com

² Center for Pattern Recognition and Machine Intelligence, Concordia University,
Montréal, Québec, Canada H3G 1M8
suen@cenparmi.concordia.ca

³ Department of Computer Science and Software Engineering,
Concordia University, 1455 de Maisonneuve Blvd. W. Montréal,
Québec, Canada H3G 1M8
krzyzak@cs.concordia.ca

Abstract. An efficient low-level word image representation plays a crucial role in general cursive word recognition. This paper proposes a novel representation scheme, where a word image can be represented as two sequences of feature vectors in two independent channels, which are extracted from vertical peak points on the upper external contour and at vertical minima on the lower external contour, respectively. A data-driven method based on support vector machine is applied to prune and group those extreme points. Our experimental results look promising and have indicated the potential of this low-level representation for complete cursive handwriting recognition.

1 Introduction

Although much progress has been made in off-line cursive word recognition over the past decade [1] [2], this task is still very challenging and the performance of the current recognition systems is far from that of human beings [3]. The solutions to several key problems related to handwritten word recognition remain unknown. One of the most important problems is the efficient low-level representation of a cursive word image for classification. Intuitively, although a handwritten word is concatenated by a small set of handwritten characters (52 characters in English) from left to right, its shape exhibits considerable variations and depends on the uncertainty of human writing. The boundaries between characters in a handwritten word are intrinsically ambiguous due to the overlapping and inter-connections. The changes in the appearance of a character usually depend on the shapes of neighboring characters (*coarticulation effects*). As a result, it is very difficult to represent the word image based on characters in early visual processing.

In the current literature several methods have been proposed to alleviate the character segmentation problem [4]. In the first case, the image of the given

word is regarded as an entity in the whole. A word is characterized by a sequence of features such as length, loops, ascenders and descenders. No sub-models are used as the part of its classification strategy. Although the method can avoid the difficult problem of segmentation completely, no uniform framework in the literature has been presented to extract those features. It is not clear how to solve the problem of the correspondence of feature points if some features are used as local shape descriptors.

In the second case, the word image is segmented into a sequence of graphemes in left-to-right order [5]. The grapheme may be one character or parts of characters. After the segmentation, possible combinations of adjacent graphemes are fed into a character recognizer. Then a dynamic programming technique is used to choose the best sequence of characters. There are two problems with this method. One is that segmentation and grapheme combination are both based on heuristic rules that are derived by human intuition. They are error-prone. The other is that the computational cost is prohibitively high due to the evaluation of a large grapheme combination.

In the third case, features are extracted in a left-to-right scan over the word by a sliding window [6]. In this method, no segmentation is required. But there are several problems related to it. One is that some topological information such as stroke continuity and contour length will be lost. But stroke continuity is a strong constraint for handwritten signals. The other is how to determine the optimal width of a sliding window. Moreover, this one-dimensional sampling of two-dimensional word image will result in information loss.

By reviewing the above methods, we know that none of them imply where the important information is located in a word image and how to organize them efficiently. In this paper, we locate certain extreme points in the vertical direction on a contour, then apply support vector machines to classify those points into two channels: local peaks in the upper external contour and local minima in the lower external contour. For classification task, local feature vectors are extracted at those points. As a result, a cursive word image will be represented by two sequences of feature vectors.

In Section 2, we discuss the relationship of feature points to the process of handwriting production and an algorithm for the extraction of those points will be given. Then we present the method of feature extraction in Section 3. The experimental results are described in Section 4. Finally, we summarize this paper and draw conclusions.

2 Locating Extreme Points

In the process of handwriting production, strokes are basic units and a handwriting signal can be represented as a sequence of strokes in the temporal dimension. A stroke is bounded by two points with curvature. In offline handwriting, the image contour can be used to precisely represent a binary (black/white) word image. The high-curvature points can be detected robustly. As a result, the external contour can be broken up into strokes under the assumption of contiguity.

In terms of an oscillatory motion model of handwriting [7], we know that the horizontal and vertical directions are more important than the other orientations. Then strokes are split into two groups: strokes in the upper contour and those in the lower contour. These strokes are ordered from left to right. This representation has several characteristics:

1. It is compatible with the study in psychology which shows that strokes are basic units of handwriting signals.
2. The space neighboring relationship of strokes is preserved.
3. It is a local representation of word image. It is easier to extract low-level invariant local features.
4. It is a 2D representation.

Also, unlike wavelet coding, more high-level units such as letters and words can be visually constructed from this representation. As a result, this representation will facilitate us in building a hierarchical system of cursive word recognition.

In order to obtain the above representation, we first need to locate those interesting points with high curvature. In the writing process, the most important part of a curve seems to be where the writing speed has reached a minimum and curvature reaches a maximum [8][9]. The vertical extrema on an external contour are the segmentation points of handwritten strokes. Fig. 1 shows the procedures of the extraction of vertical extrema. The interesting locations are the peaks in the upper external contour and minima in the lower external contour. The neighboring curved segments at those extrema are convex. If the curve are smooth, the curvatures at those points are positive. Theoretically there exists

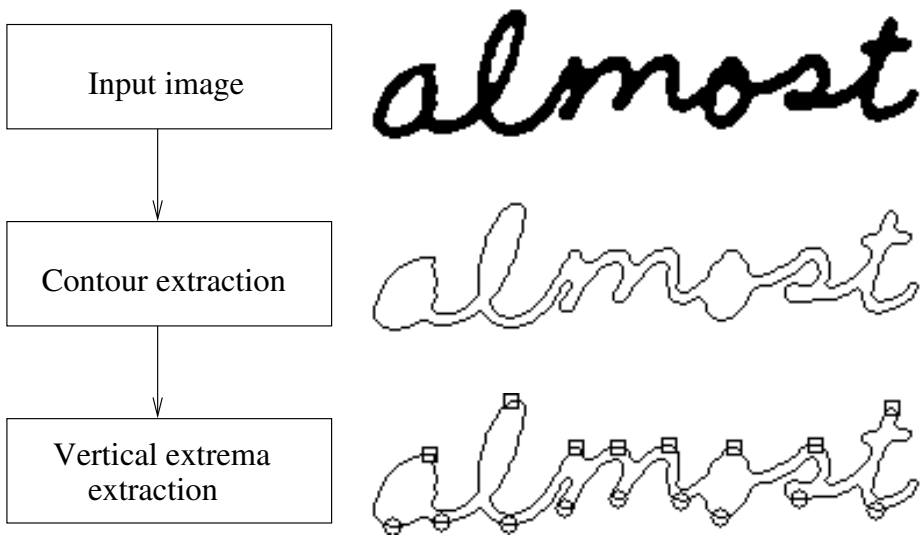


Fig. 1. Vertical extrema on the external contour of a word image. The peaks in the upper contour are marked by rectangles. The minima in the lower contour are marked by circles

a point with the negative curvature between two neighboring extrema. This indicates that a point with the negative curvature depends on its neighboring points with positive curvature. Therefore, it is reasonable to assume that peaks in the upper external contours are pairwise independent. So are the minima in the lower external contours. Also, these locations are analogous to the centers of receptive fields [10], which attract visual attention [11]. Separating these extrema into two groups have the following advantages:

- 2D space configuration of these extrema can be approximated by two 1D space configurations. Consequently the problem complexity will be greatly reduced.
- It conforms with the Gestalt principles of perceptual organization: proximity principle (vertical distances) and similarity (local curve shape at these points).
- When we model the signal similarity independently and signals in one group are degraded, the model in the other group is not affected.

In addition, for the inner contour, we represent it as a loop in the stage of feature extraction, rather than as vertical extrema. The loop will be associated with the closest extrema in the external contour. In the next stage, local features are extracted at those extrema.

Vertical extrema can be detected on the image external contour. We propose an algorithm to detect those points robustly as below:

Algorithm for Detection of Vertical Extrema

Input: contour points $\mathbf{v}[i]$, $i = 1, \dots, n$ and working space \mathbf{v}_2 and \mathbf{v}_3 .

Output: peaks and minima

- 1 Identify the index set $B = \{i | \mathbf{v}[i].y \neq \mathbf{v}[i + 1].y\}$.
- 2 Copy elements in set B into the vector \mathbf{v}_2 whose elements are sorted in an increasing order. The length of vector is $K = |B|$.
- 3 Calculate the difference of y-coordinate of two neighboring points in \mathbf{v}_2 .
 $\mathbf{v}_3[k] \leftarrow \text{sign}(\mathbf{v}[\mathbf{v}_2[k]].y - \mathbf{v}[\mathbf{v}_2[k] + 1].y)$, $k = 1, \dots, K$.
- 4 Median filtering of window size 3 is applied to \mathbf{v}_3 .
- 5 Select the candidates of extrema from the two indexed vectors:
 $\text{peak}[i] \leftarrow \mathbf{v}_2[k]$ if $\mathbf{v}_3[k] < 0$; $\text{minima}[j] \leftarrow \mathbf{v}_2[k]$ if $\mathbf{v}_3[k] > 0$. P and M denote the number of peaks and number of minima, respectively.
- 6 Prune invalid minima iteratively.
- 7 Prune invalid peaks iteratively.

In the above algorithm, three primary measures are used to prune invalid peaks: contour length, height difference and bounded variation between two neighboring peaks. For example, if the contour length between two neighboring peaks is small, they will be merged into one peak that will be located at the middle point on the contour. Also, if a local minimum point is located in the upper zone, it will be pruned.

3 Features

Although the algorithm in Section 2 can be applied to group peaks and minima and prune invalid extrema, it is still not good enough due to various variations of word shapes. Therefore, we need to introduce a classifier to refine the grouping and pruning process. Several features have to be extracted at each extreme point. We describe these features as follows:

1. Number of local minima on the current contour (f_1). In Fig. 2, the feature values at points 1 and 2 are 2.
2. Number of local peaks on the current contour (f_2). In Fig. 2, the feature values at points 1 and 2 are 2 and 1, respectively.
3. Minimum height difference with neighboring extrema (f_3). When the current point is a local minimum, two neighbors are local minima; When the current point is local peak, two neighbors are local peaks. Neighbors may not be on the same contour as the current one. In Fig. 3, the feature value at point 2 is $\min(|y_1 - y_2|, |y_3 - y_2|)$.
4. Minimum height difference with neighboring extrema (f_4). When the current point is a local minimum, two neighbors are local peaks; when the current

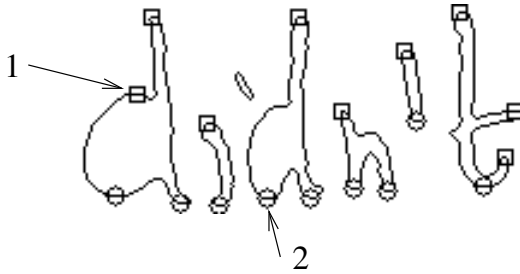


Fig. 2. Illustration for the extraction of number of peaks and minima

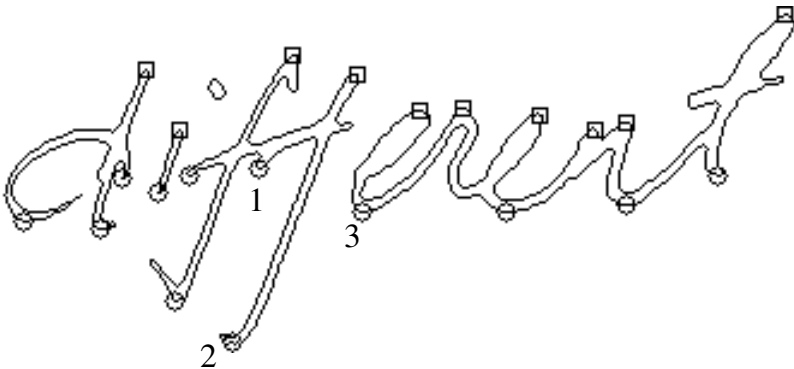


Fig. 3. Illustration for the extraction of minimum height difference. The neighbors have the same convex attributes as the current extreme point

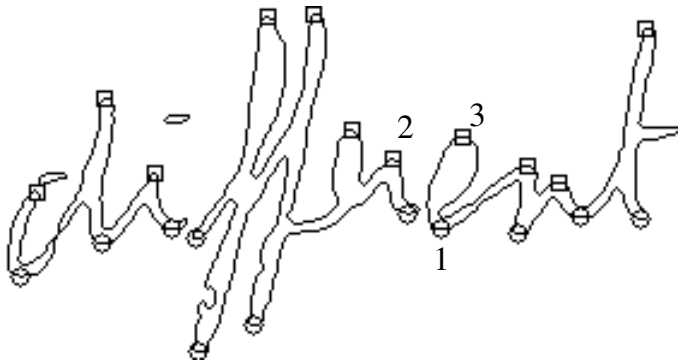


Fig. 4. Illustration for the extraction of minimum height difference. The neighbors have different convex attribute from the current extreme point

point is local peak, two neighbors are local minima; Neighbors may not be on the same contour as the current one. In Fig. 4, two neighbors of point 1 are 2 and 3. The feature value at point 1 is $\min(|y_1 - y_2|, |y_3 - y_1|)$

5. Percent of peaks above the current point (f_5).
6. Percent of minima below the current point (f_6).
7. Relative position in y axis (f_7). The feature is defined by $\frac{y}{h}$, where y is the coordinate of the current point in y-axis and h is the image height.
8. Minimum contour length with neighboring peaks (f_8). Note that neighboring peaks are on the same contour as the current point.
9. Minimum contour length with neighboring minima (f_9). Note that neighboring minima are on the same contour as the current point.
10. Minimum height difference with neighboring peaks (f_{10}). The neighboring peaks must be on the same contour as the current point.
11. Minimum height difference with neighboring minima (f_{11}). The neighboring minima must be on the same contour as the current point.

In the above features, f_1 and f_2 characterizes the information of word length. f_3 represents the information of ascender and descender. For each feature value, a reasonable upper bound will be set. If the feature value is greater than the corresponding bound, it will be rounded to this bound. In biological learning, it is called “peak effect” [12]. Given that the specified bound b_i of the feature f_i , the round operation is given by

$$f'_i = \min(f_i, b_i) \tag{1}$$

For fast learning, the feature values will be first transformed into the interval $[0, 1]$, the variable transformation $x^{0.4}$ is applied to each component of the feature vector such that the distribution of each feature is Gaussian-like [13]. The formula is given by

$$f''_i = \left(\frac{f'_i}{b_i}\right)^{0.4} \tag{2}$$

Then a support vector classifier [14] is constructed to classify points to two classes: extrema on the upper contour and extrema on the lower contour. If one local minimum is classified to the upper contour, it will be pruned. If one local peak is classified to the lower contour, it will be pruned. As a result, the valid local peaks will be on the upper contour while the valid local minima will be on the lower contour.

4 Experiments and Results

The word representation method described in this paper has been implemented in C++ on Windows XP and compiled by Microsoft visual C++6.0. It is a part of IMDS handwritten word recognition engine. The experiments were conducted on IMDS cursive word database. At IMDS Software we designed a specified electronic form to collect isolated handwritten words such that labelling can be done automatically. Presently the vocabulary are the words from the category of Collins Frequency Band 5, in which these words are most frequently used in daily life. The size of this lexicon is 670. Our samples are written by a variety of 78 persons from different countries such as Arabian, Asian, Canadian, French., from students and professors at universities, employees in companies. No constraints are imposed on the writers in order to get most natural handwritten samples. Each writer writes samples in blank boxes in the form, which contains 670 words. This indicates there are no two samples from the same writer for each word. The form is scanned as a gray-scale image in 300 DPI and is binarized. The samples are randomly split into training and testing sets whose sizes are 38795 and 13733, respectively. Some samples are depicted in Fig. 5.

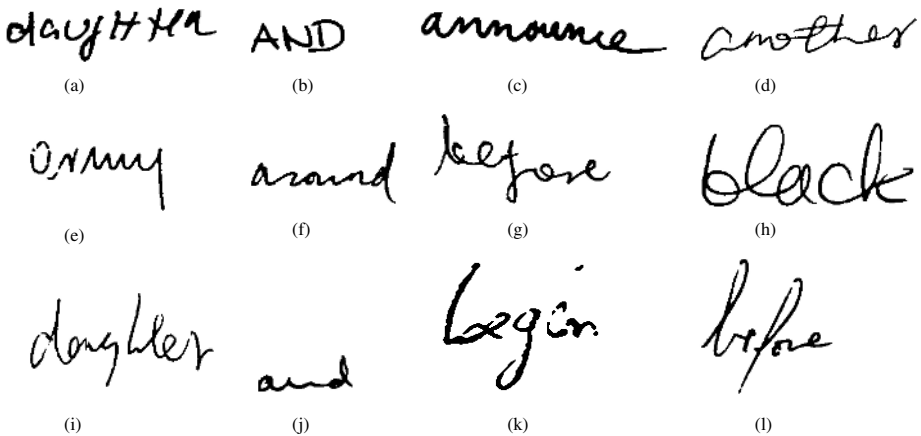


Fig. 5. Some samples in IMDS cursive word database: (a) “daughter”, (b) “and”, (c) “announce”, (d) “another”, (e) “army”, (f) “around”, (g) “before”, (h) “black”, (i) “daughter”, (j) “and”, (k) “begin”, and (l) “before”

Table 1. Upper bounds $b_i, i = 1, \dots, 11$

b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}
13.0	13.0	70.0	70.0	1.0	1.0	1.0	120.0	120	70.0	70.0

Table 2. Performance of support vector machine

Training set	Testing set	Training rate	Testing rate	SV	BSV
354,928	119,825	99.78%	99.6%	4377	2936

Support vector machine (SVM) is used as a classifier to classify extrema into two categories. The radial basis function (RBF) is chosen as the SVM kernel, given by

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{0.6^2}\right) \quad (3)$$

where $\|\cdot\|$ denotes Euclidean norm and the dimension of feature vectors x and x' is 11. The upper bounds $b_i, i = 1, \dots, 11$ of eleven features in Section 3 are shown in Table 1. The value of C in the dual form of SVM [14] is set to 10.0. SVM is trained by HeroSvm2¹. Some labelled samples have to be collected before we train support vector machine. Our strategy is first to label a small number of extrema in the word images manually. Then these samples are divided into training and testing sets. A SVM classifier is constructed. The classifier is used to label other extrema. The misclassified errors are corrected manually. Since the recognition rate of SVM classifier is very high, more than 99%, the number of manual corrections is small. Much time-consuming cost has been saved. Table 2 shows the performance of SVM classifier, where SV and BSV denote the number of support vectors and number of bounded support vectors, respectively. The number of support vectors is small, compared with the size of the whole training set. It may indicate that features are discriminative so that a small portion of SVs can characterize the classification boundary. The above results look very promising. They indicate that the extrema can be grouped into two categories with a high accuracy though cursive word shapes exhibit considerable variations. It also infers that in low-level visual processing the data-driven learning technique with top-down information can eliminate the uncertainty of a decision to a great extent. Traditionally, the baseline information may be used to determine the upper and lower zones. But the detection of baseline is not robust due to uneven writing. Moreover, it is difficult to find a baseline for some short words. One of the appealing properties of the proposed method is that the output of SVM classifier can be used as the confidence value. When the absolute value of SVM's output is larger, the decision of the classifier becomes more reliable. Some examples are shown in Fig. 6. It can be observed that the proposed method is insensitive to the word length and image size scale.

¹ <http://www.cenparmi.concordia.ca/people/jdong/HeroSvm.html>.

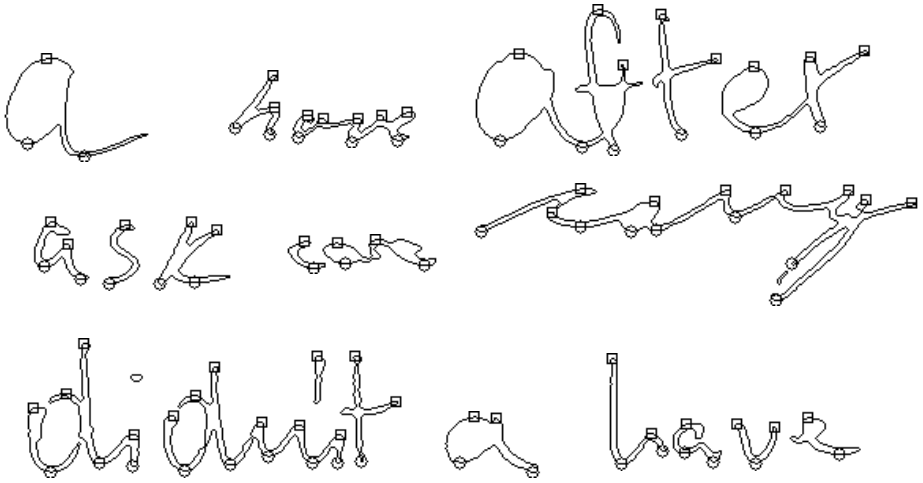


Fig. 6. Some word images where extrema are classified to upper peaks and lower minima. Peaks and minima are marked by squares and circles, respectively

5 Conclusions

In this paper, we present an efficient low-level representation of cursive word images for classification task, which is an important step to build a general hierarchical word recognition system. A word image can be represented as two sequences of feature vectors which are extracted at vertical peak points on the upper contour and at vertical minima on the lower contour. Some evidences from the process of handwriting production and visual perception are linked to this representation. The experimental results look promising and have shown the potential of this representation for cursive word recognition task, which is our primary goal.

Acknowledgments

This research is support by an Industrial Research Fellowship awarded by Natural Sciences and Engineering Research Council of Canada and IMDS Software.

References

1. Steinherz, T., Rivlin, E., Intrator, N.: Offline cursive script word recognition – a survey. *International Journal on Document Analysis and Recognition* **2** (1999) 90–110
2. Koerich, A., Sabourin, R., Suen, C.: Large vocabulary off-line handwriting recognition: A survey. *Pattern Analysis and Applications* **6** (2003) 97–121

3. Bunke, H.: Recognition of cursive roman handwriting – past, present and future. In: Proceedings of IEEE 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland (2003) 448–459
4. Madhvanath, S., Govindaraju, V.: The role of holistic paradigms in handwritten word recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23** (2001) 149–164
5. Knerr, S., Augustin, E., Baret, O., Price, D.: Hidden Markov Model based word recognition and its application to legal amount reading on French checks. *Computer Vision and Image Processing* **70** (1998) 404–419
6. Mohamed, M., Gader, P.: Handwritten word recognition using segmentation-free hidden markov modeling and segmentation-based dynamic programming techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18** (1996) 548–554
7. Hollerbach, J.: An oscillation theory of handwriting. *Biological Cybernetics* **39** (1981) 139–156
8. Attneave, F.: Some informational aspects of visual perception. *Psychological Review* **61** (1954) 183–193
9. Schomaker, L., Segers, E.: Finding features used in the human reading of cursive handwriting. *International Journal on Document Analysis and Recognition* **2** (1999) 13–18
10. Hartline, H.: The response of single optic nerve fibres of the vertebrate eye to illumination of the retina. *American Journal of Physiology* **121** (1938) 400–415
11. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20** (1998) 1254–1259
12. Staddon, J.: *Adaptive Behavior and Learning*. Cambridge University Press, Cambridge (1983)
13. Fukunaga, K.: *Introduction to Statistical Pattern Recognition*. second edn. Academic Press (1990)
14. Vapnik, V.: *Statistical Learning Theory*. Wiley, New York (1998)

Supervised Evaluation of Dataset Partitions: Advantages and Practice

Sylvain Ferrandiz^{1,2} and Marc Boullé¹

¹ France Télécom R&D, 2, avenue Pierre Marzin,
22307 LANNION Cedex, France

² Université de Caen, GREYC, Campus Côte de Nacre,
boulevard du Maréchal Juin, BP 5186,
14032 Caen Cedex, France

{sylvain.ferrandiz, marc.boullé}@francetelecom.com

Abstract. In the context of large databases, data preparation takes a greater importance : instances and explanatory attributes have to be carefully selected. In supervised learning, instances partitioning techniques have been developed for univariate representations, leading to precise and comprehensible evaluations of the amount of information contained in an attribute, with respect to the target attribute. Still, the multivariate case remains unstated.

In this paper, we describe the partitioning intrinsic convenience for data preparation and we settle a framework for supervised partitioning. A new evaluation criterion of labelled objects partitions, which is based on Minimum Description Length principle, is then set and tested on real and synthetic data sets.

1 Supervised Partitioning Problems in Data Preparation

In a data mining project, the data preparation phase is a key one. Its main goal is to provide a clean and representative database for the consecutive modelling phase [3]. Typically, topics like instances representation, instances selection and/or aggregation, missing values handling, attributes selection, are to be carefully dealt with. Among the many designed methods, partition-based one are often used, for their ability to comprehensibly summarize the information.

The first examples that come in mind are clustering techniques, like the most popular one : K -means [11], which aim at partitioning instances. Building partitions hierarchy or mixture models is another way of doing unsupervised classification [5]. Combining clustering and attributes selection has led to the description of self-organizing feature maps [10].

In the supervised context, induction tree models are plainly partition-based [2],[12],[8]. These models build a hierarchy of instances groups relying on the discriminating power of the explanatory attributes with respect to the categorical target attribute. As the naive Bayes classifier, they need to discretise the continuous explanatory attributes to make probability estimations more accurate. As

discretisation is the typical univariate supervised partitioning problem, we now take a closer look at it.

The objective of the discretisation of a single continuous explanatory attribute is to find a partition of the values of this attribute which best discriminates the target distributions between groups. These groups are intervals and the partition evaluation is based on a compromise : fewer intervals and stronger target discrimination are better. There are mainly two families of search algorithms : bottom-up greedy agglomerative heuristics and top-down greedy divisive ones.

Discrimination can be evaluated in four ways using statistical test, entropy, description length or bayesian prior :

- Chimerge [9] applies chi square measure to test the independance of the distributions between groups,
- C4.5 [12] uses Shannon’s entropy based information measures to find the most informative partition,
- MDLPC [6] defines a description length measure, following the Minimum Description Length principle [13],
- MODL [1] states a prior probability distribution, leading to a bayesian evaluation of the partitions.

The discretisation problem is illustrative of the convenience of supervised partitioning methods for data preparation since it addresses simultaneously the three following problems :

- Data representation : a suitable representation of the objects at hand have to be selected. Partitioning is an efficient mean to evaluate representations quality (in the supervised context, statistical test for class separability is another one, cf. [14]).

Table 1. Examples of resulting partitions of Fisher’s Iris database for different representation spaces. Partitioning techniques allow, among other things, to carry out the selection of an attribute subset in an intelligible way, as the results are quickly interpretable and easily comparable. Here, we see that the three iris categories (Setosa, Versicolor and Virginica) are completely discrimated by the four attributes. However, one can consider petal width only. Furthermore, one can state that setosas distinguish themselves by their sepal width and length

	Labels distributions in groups								
	Group 1			Group 2			Group 3		
Explanatory attributes	Set.	Ver.	Vir.	Set.	Ver.	Vir.	Set.	Ver.	Vir.
Sepal width, sepal length, petal width, petal length	50	0	0	0	50	0	0	0	50
Petal width, petal length	50	0	0	0	50	1	0	0	49
Sepal width, sepal length	50	2	1	0	48	49			
Petal width	50	0	0	0	48	0	0	2	50

- Interpretability : labelled groups result from an understandable compromise between partition simplicity and target discrimination.
- Comparison capacity : explanatory attributes effects on the target can be quickly compared.

These themes are intertwined and play a crucial role in the data preparation phase (cf. Table 1 for an intuitive illustration in the multivariate case). The goal of this paper is to set a framework for supervised partitioning and to specify an evaluation criterion, preserving the interpretability bias and allowing not to consider single continuous attributes only.

In the remainder of the paper, we first set our framework and a description method of partitions (section 2). Then, we propose a new evaluation criterion (section 3) and we test its validity on real and synthetic datasets (section 4). Finally, we conclude and point out future works (section 5).

2 Graph Constrained Supervised Partitioning

Let $O = \{o_1, \dots, o_N\}$ be a finite set of objects. A target l_n lying in an alphabet of size J is associated to each object o_n and a graph structure G is set on O . This structure can be natural (road networks, web graphs, ...) or imposed (proximity graphs, partial orders, ...). In the remainder, we will suppose G non-oriented. Our problem consists in finding an optimal partition of G , considering partitions composed of connected groups with respect to the discrete structure (i.e *connected partitions*). As explained above, optimality of a partition relies on the correct balance between the structure of its groups and its discriminating power (cf Figure 1). The setting of the balance requires the definition of description parameters both for the structure and the target distribution.

Let π be a connected partition of G . We now introduce an effective and interpretable bias. We consider the balls induced by the discrete metric $\delta : \delta(o_1, o_2)$ is the minimum number of edges needed to link o_1 and o_2 . As illustrated by Figure 2, each group of π is then covered with δ -balls.

The method consists in selecting non-intersecting balls that are included in a group of π . At each step, the biggest one is picked up :

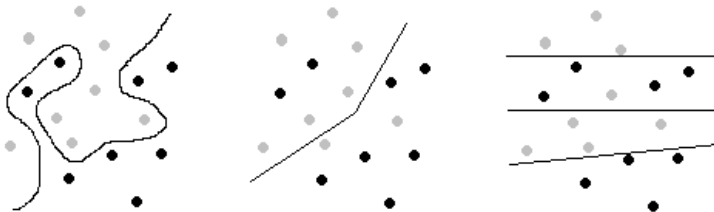


Fig. 1. 2 classes problem: which is the "best" partition?

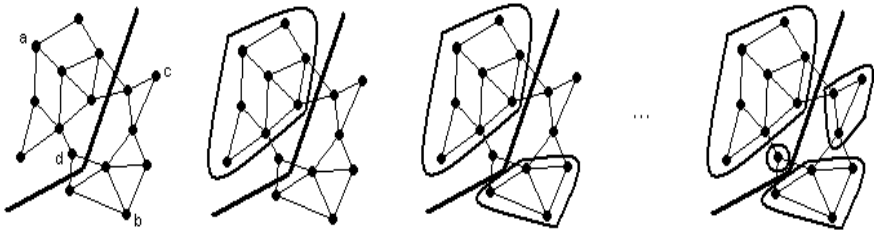


Fig. 2. Applying algorithm 1: description of a partition with non-intersecting balls ($B(a, 2), B(b, 1), B(c, 1), B(d, 0)$) defined by the graph distance

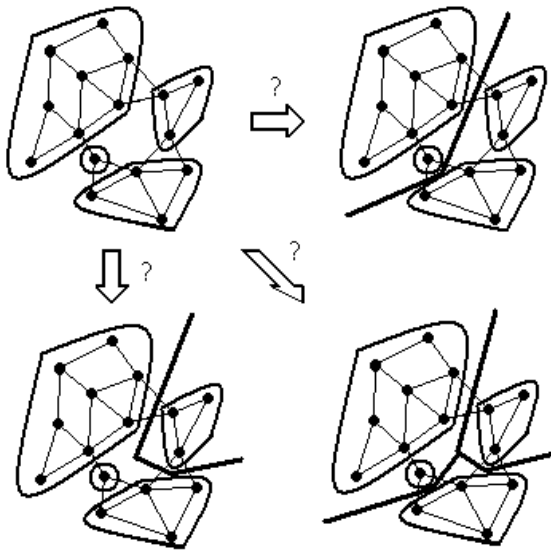


Fig. 3. Examples of possible partitions obtained with different groupings of the balls

Algorithm 1 :

- $A \leftarrow O$
- $B \leftarrow \emptyset$
- **While** $A \neq \emptyset$ **Do**
 - $S \leftarrow$ the ball with maximal size included in A and in a group of π
 - $B \leftarrow B \cup \{S\}$
 - $A \leftarrow A \setminus S$

However, the set B does not characterise π : different partitions can give the same set B (cf Figure 3). But if the number of groups K is considered as a description parameter, obtaining π from B is the same as putting these balls in

K different boxes. Finally, π is fully described by the set of balls B , the number K and a partition of B in K groups. This is not a compact description as we do not take into account the graph structure in the second step. Indeed, some partitions of B in K groups do not lead to connected partitions and should not be taken into account.

The description parameters of the target distribution are more easily caught. If π_k is one of the K groups in π , describing its inner labels distribution is the same as putting the objects contained in π_k in J boxes. This is done by firstly assigning the numbers N_{kj} of objects in π_k to put in the j^{th} box, and secondly specifying the partition of the group π_k in J groups of sizes N_{k1}, \dots, N_{kJ} .

The description bias allows to define the *structural complexity* of π relying on its ball decomposition in an interpretable way : fewer and bigger balls means simpler structure. The description of the target distribution in terms of frequency parameters leads to an informational definition of the *target discrimination* : strong discrimination is related to low entropy. The evaluation of a partition must result from a compromise as strong discrimination goes with high structural complexity.

3 Evaluation Criterion

Let π be a connected partition of O . To set an evaluation criterion $l(\pi)$, the Minimum Description Length principle is applied [13], for its intrinsic ability to handle compromises. The problem turns into a two-step description problem : description of the parameters defining groups and description of the labelling parameters. This leads to write

$$l(\pi) := l_{structure}(\pi) + l_{labels/structure}(\pi),$$

with l standing for description lengths function. A description protocol must be designed from which description lengths can be specified.

As the structure is characterised by a set of balls B and a partition of B , its description length is split into the sum of the description length of B and that of the partition of B :

$$l_{structure}(\pi) := l_{ballset}(\pi) + l_{ballsgrouping}(\pi).$$

As the distributions are characterised in each group by the frequencies of the labels and a partition, the related description length is split into the following way :

$$l_{labels/structure}(\pi) := \sum_{k=1}^K l_{frequencies}(\pi_k) + \sum_{k=1}^K l_{partitioning/frequencies}(\pi_k),$$

where $\pi = (\pi_1, \dots, \pi_K)$.

In the first place, let's form a description protocol of the balls set B . The balls in B are ordered by decreasing sizes $d_1 > \dots > d_p$ and if t_i is the number

of balls of size d_i , B_j^i refers to the j^{th} ball ($1 \leq j \leq t_i$) of size d_i ($1 \leq i \leq p$). The protocol consists in specifying by decreasing size which is the next ball of the description and, when needed, the next size to be considered. Precisely :

- $p \leftarrow 1$
- describe d_p
- **if** $d_p < N$ **then**
 - **While** $d_p > 1$
 - * describe successively $B_1^p, \dots, B_{t_p}^p$
 - * $p \leftarrow p + 1$
 - * describe d_p

As description lengths can be interpreted as negative log of probabilities, we just have to assign probabilities to obtain $l_{\text{ballset}}(\pi)$. We choose a uniform prior for each parameter description step and description lengths are computed using counting. For example, description length of d_1 is $\log_2(N)$ as possible values of d_1 are $1, \dots, N$. Description length of d_2 is $\log_2(d_1 - 1)$ as, at this step, the possible values of d_2 are $1, \dots, d_1$ and so on. Besides, the description length of B_1^1 is $\log_2 \beta_1^1$, where β_1^1 is the total number of balls of size d_1 induced by the discrete structure G . That of B_2^1 is $\log_2 \beta_2^1$, where β_2^1 stands for the total number of balls of size d_1 induced by G that do not intersect B_1^1 , etc... The overall sum of these lengths defines $l_{\text{ballset}}(\pi)$.

In the second place, to set $l_{\text{ballgrouping}}(\pi)$, we describe the group number K of π and the partition of B in K groups. Once again, a uniform prior is applied. As K lies between 1 and the size K_B of B and as the number of partitions of B in less than K groups is $B(K_B, K)$ (the sum of the K first Stirling numbers), we have

$$l_{\text{ballgrouping}}(\pi) = \log_2 K_B + \log_2 B(K_B, K).$$

In the third and final place, applying a uniform prior to obtain the description lengths of the target leads to set

$$l_{\text{labels/structure}}(\pi) = \sum_{k=1}^K \log_2 \binom{N_k + J - 1}{J - 1} + \sum_{k=1}^K \log_2 \frac{N_k!}{N_{k1}! \dots N_{kJ}!}.$$

The first sum results from the description of the labels frequencies (N_{k1}, \dots, N_{kJ}) in each group k . These J -tuples satisfy the property $\sum N_{kj} = N_k$ (with N_k the size of group k), and $\binom{N_k + J - 1}{J - 1}$ is the number of such tuples. The number of partitions of a set of size N_k in J groups of sizes N_{k1}, \dots, N_{kJ} is the multinomial coefficient $\frac{N_k!}{N_{k1}! \dots N_{kJ}!}$. That gives the second sum.

4 Experiments

The experiments are performed using the standard hierarchical greedy bottom-up heuristic, the initial partition being that with one object per group :

Algorithm 2.

- $\pi \leftarrow \text{InitialPartition}$
- **For** $k = 2$ to N **Do**
 - $\pi \leftarrow$ the best partition resulting from the merging of two groups of π
- **Return** the overall best partition encountered

Thus, $O(N^2)$ partitions are evaluated. The greedy character of this heuristic does not allow to evaluate a significant part of the partitions set and such a method easily falls into local optima. To alleviate these facts, we select a more appropriate initial partition : initial groups are the biggest clean balls (i.e objects in a ball share the same label).

A graph structure has to be selected. As the objects are always imbedded in an euclidean space, the experiments are carried out with the Gabriel graph, which is a proximity graph [7]. The distance between two objects o_1 and o_2 is taken to be the imbedding euclidean one L and these objects are adjacent in the Gabriel sense (cf Figure 4) if and only if

$$L(o_1, o_2)^2 \leq \min_{o \in O} L(o_1, o)^2 + L(o_2, o)^2.$$

We perform two experiments on synthetic datasets and one on real datasets. In a first one, we check the criterion ability to detect the independence between the descriptive attributes and the target one, on synthetic datasets. These are two-classes problems, with points uniformly generated inside the Hamming hypercube and each point label uniformly assigned. The varying parameters are the number N (from 1 to 100) of points and the space dimension d (taking values 1, 2, 3, 5 and 10). For each couple of values, 25 datasets are generated. For every dataset, our method builds a partition composed of one single group. This is exactly the expected behavior : no discrimination has to be done since the target is independent of the explanatory attributes.

In a second experiment, we test the criterion discrimination ability for gaussian mixture models in the plane. We settle a four gaussians problem, centered in

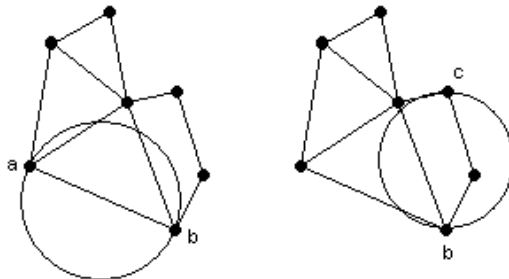


Fig. 4. Example of a Gabriel graph. The ball of diameter $[ab]$ contains no other point : a and b are Gabriel-adjacent. The ball of diameter $[bc]$ contains another point : b and c are not Gabriel-adjacent

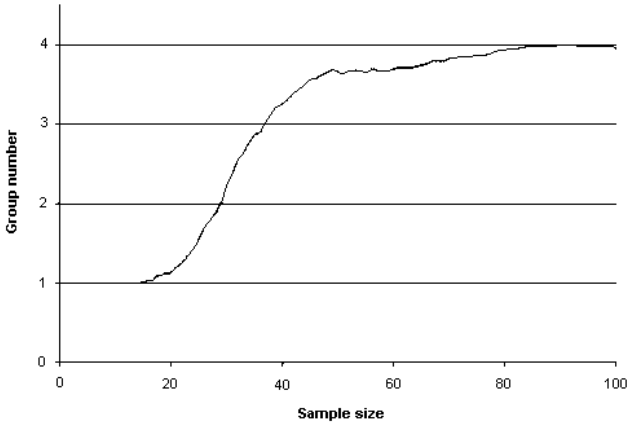


Fig. 5. Resulting number of groups on the four gaussians problem

Table 2. Prediction accuracy of both NN and partition-based rules and group number of the partition. Our method gives additional information : each class lies in a single cluster, for every datasets

Dataset	NN	Partition	Group number
Iris	0.95 ± 0.03	0.94 ± 0.04	3 ± 0.0
Wine	0.95 ± 0.03	0.94 ± 0.04	3 ± 0.0
Breast	0.96 ± 0.01	0.96 ± 0.01	2 ± 0.0

$(1, 1)$, $(-1, 1)$, $(-1, -1)$ and $(1, -1)$, with diagonal covariance matrix $\begin{pmatrix} 1/4 & 0 \\ 0 & 1/4 \end{pmatrix}$. The varying parameter is the number N of points and for each value, 25 datasets were generated. Figure 5 shows that, with sufficiently many points, the four groups are detected. The detection threshold could however be better. Indeed, as we do not take into account the graph structure for the description of the balls set partition, the description length $l_{ballsgrouping}$ is over-estimated. To obtain a decrease of the total description length, the (too big) increasing of the structural length induced by the decision of creating a new group must be balanced by a (very) strong resulting discrimination.

In a third experiment, we consider the resulting partition as a predictive model : a new instance is classified according to a majority vote in the nearest group. The evaluation consists in a stratified five-fold cross-validation and results of the Nearest Neighbor (NN) rule [4] are given for comparison. The tests were carried through 3 datasets from the UCI machine learning database repository.

The well-known Fisher's Iris dataset contains 150 instances of such flowers described by 4 continuous explanatory attributes and belonging to one of the three classes Setosa, Versicolor and Virginica (as previously seen). The Wine database results from the analysis of 13 components found in each of 3 types of wines, and is composed of 178 instances. Finally, the Breast dataset aims at

studying the malignant character of a breast cancer for 699 subjects through 9 descriptive attributes.

In order to limit scale effects on the distance measure, each explanatory attribute is linearly transformed to lie in $[0, 1]$. Table 2 summarizes the results of the evaluation. The main advantage of partitioning methods lies in the fact that they detect or supply an underlying structure of the analysed data. As this structural gain may be balanced by an information loss, it's noteworthy that, on the three datasets, our technique does not suffer from such a curse.

5 Conclusion and Further Works

In this paper, we have discussed the usefulness of supervised partitioning methods for data preparation, set a framework for supervised partitioning, proposed and tested an evaluation criterion of labelled partition. The representation quality of the objects and their inner amount of information about the target attribute can be subtly and simply evaluated, whatever may be the kind of the objects. Specifically, multivariate representations can be considered.

The proposed method builds an underlying structure of the data : a partition. This is done in an understandable way (with the use of balls) and without loss of predictive information (as shown by the experiments on real datasets). The settled criterion is able to detect independence too. If the explanatory attributes contain no information with respect to the target attribute, the "best" partition should be that with one group and that's the way the criterion behaves.

Still, this is preliminary work. The presented criterion can be improved. The "balls grouping" description part could take into account the graph structure, leading to a more accurate evaluation criterion.

As well, the greedy agglomerative approach is not effective and easily falls into a local optimum. Furthermore, the heuristic lacks of computational efficiency : the complexity's polynomial order is too high for real applications. In future works, we plan to design a heuristic founded on the description bias (the use of balls).

References

- [1] Boullé, M.: A bayesian approach for supervised discretization. *Data Mining V*, Zanasi and Ebecken and Brebbia, WIT Press (2004) 199–208
- [2] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: *Classification and regression trees*. California: Wadsworth International (1984)
- [3] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: *CRISP-DM 1.0 : step-by-step data mining guide*. Applied Statistics Algorithms (2000)
- [4] Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory* **13** (1967) 21–27
- [5] Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern classification*. Wiley-Interscience (2000)

- [6] Fayyad, U., Irani, K.: On the handling of continuous-valued attributes in decision tree generation. *Machine Learning* **8** (1992) 87–102
- [7] Jaromczyk, J.W., Toussaint, G.T.: Relative neighborhood graphs and their relatives. *P-IEEE* **80** (1992) 1502–1517
- [8] Kass, G.: An exploratory technic for investigating large quantities of categorical data. *Journal of Applied Statistics* **29** (1980) 119–127
- [9] Kerber, R.: Chimerge discretization of numeric attributes. Tenth International Conference on Artificial Intelligence (1991) 123–128
- [10] Kohonen, T.: Self-organized formation of topologically correct feature maps. *Biological Cybernetics* **43** (1982) 59–69
- [11] McQueen, J.: Some methods for classification and analysis of multivariate observations. Fifth Berkeley Symposium on Mathematical Statistics and Probability, Le Cam and Neyman **1** (1967) 281–297
- [12] Quinlan, J.R.: *C4.5: Programs for machine learning*. Morgan Kaufmann (1993)
- [13] Rissanen, J.: Modeling by shortest data description. *Automatica* **14** (1978) 465–471
- [14] Zighed, D.A., Lallich, S., Muhlenbach, F.: Séparabilité des classes dans \mathbb{R}^p . VIIIème Congrès de la Société Francophone de Classification (2001) 356–363

Inference on Distributed Data Clustering

Josenildo C. da Silva* and Matthias Klusch

German Research Center for Artificial Intelligence,
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
{jcsilva, klusch}@dfki.de

Abstract. In this paper we address confidentiality issues in distributed data clustering, particularly the inference problem. We present a measure of inference risk as a function of reconstruction precision and number of colluders in a distributed data mining group. We also present KDEC-S, which is a distributed clustering algorithm designed to provide mining results while preserving confidentiality of original data. The underlying idea of our algorithm is to use an approximation of density estimation such that it is not possible to reconstruct the original data with better probability than some given level.

1 Introduction

Data Clustering is a descriptive data mining task aiming to partition a data set into groups such that data objects in one group are similar to each other and are so different as possible from those in other groups. In distributed data clustering (DDC) the data objects are distributed among several sites. The traditional solution to (homogeneous) DDC problem is to collect all the distributed data sets into one centralized repository where the clustering of their union is computed and transmitted back to the sites.

This approach, however, may be impractical due to constraints on network bandwidth or secrecy issues, when the sites are not allowed to share data due to legal issues or because it is against some local security policy. Examples of such confidential data include medical information and marketing secrets. The main problem is that confidential information may be reconstructed even if it is not explicitly exchanged among the peers. This problem, known as *inference problem*, was first studied in statistical data bases and more recently has attracted the attention of the data mining community [7].

In this paper we address the problem of homogeneous DDC considering confidentiality issues, particularly the inference problem. Informally, the problem is to find clusters using distributed set of data ensuring that, at the end of the computation, each peer only knows his own dataset and the resulting cluster mapping. Our main objective is to propose a measure of how confidential one algorithm

* This work is partially supported by CAPES (Coord. de Aperfeicoamento do Pessoal de Nivel Superior) of Ministry for Education of Brazil, under Grant No. 0791/024.

is, i.e. how vulnerable it is to inference attacks. Additionally, we present a distributed algorithm for confidential DDC with an analysis of its confidentiality level.

The remaining of this paper is organized as follows. In section 2 we present our definitions of confidentiality and inference risk. In section 3 we present our algorithm. Related work are presented in section 4. Conclusions and remarks are presented in section 5.

2 Confidentiality in Distributed Data Clustering

We define the problem of confidential distributed data clustering as follows.

Definition 1. Let $\mathcal{L} = \{L^j | 1 \leq j \leq P\}$ be a group of peers sites, each of them with a local set of data objects $D^j = \{\mathbf{x}_i | i = 1, \dots, N\} \subset \mathbb{R}^n$, with $\mathbf{x}_i^{(d)}$ denoting the d -th component of \mathbf{x}_i . Let \mathcal{A} be some DDC algorithm executed by the members of \mathcal{L} . We say that \mathcal{A} is a Confidential DDC algorithm if the following holds: (a) \mathcal{A} produce correct results (b) at the end of the computation L^j knows only the cluster mapping and its own data set D^j , with $1 \leq j \leq P$.

Our objective in this paper is to analyze how much a given distributed clustering algorithm copes with the second requirement.

2.1 Confidentiality Measure

Our starting point is the definition of a confidentiality measure. One way to measure how much confidentiality an algorithm preserves, is to ask how close one attacker can get from the original data objects. In the following we define the notion of confidentiality of data w.r.t. reconstruction. Considering multidimensional data objects, we have to look at each dimension at time.

Definition 2. Let \mathcal{L} be a group of peer as in definition 1. Let \mathcal{A} be some DDC algorithm executed by the members of \mathcal{L} . Denote by $R^k \subset \mathbb{R}^n$ a set of reconstructed data objects owned by some malicious peer L^k after the computation of the data mining algorithm, such that each \mathbf{r}_i is a reconstructed version of \mathbf{x}_i . We define the confidence level of \mathcal{A} with respect to dimension d as:

$$Conf_{\mathcal{A}}^{(d)} = \min\{|\mathbf{x}_i^{(d)} - \mathbf{r}_i^{(d)}| : \mathbf{x}_i \in D^j, \mathbf{r}_i \in R^k, 1 \leq i \leq |D^j|\}$$

Definition 3. We define the confidentiality level associated to some algorithm \mathcal{A} , as:

$$Conf_{\mathcal{A}} = \min\{Conf_{\mathcal{A}}^{(d)} | 1 < d < n\}$$

Roughly speaking, our confidentiality measure, indicates the precision with which a data object \mathbf{x}_i can be reconstructed.

In a distributed algorithm we have to consider the possibility of two or more peers forming a collusion group to disclose information owned by others. The next definition extends the confidentiality level to include this case.

Definition 4. Let \mathcal{A} be a distributed data mining algorithm. We define the function $Conf_{\mathcal{A}} : \mathbb{N} \rightarrow \mathbb{R}_+ \cup \{0\}$, representing $Conf_{\mathcal{A}}$ when c peers collude.

Definition 5 (Inference Risk Level). Let \mathcal{A} be a DDC algorithm being executed by a group \mathcal{L} with p peers, where c peers in \mathcal{L} forms a collusion group. Then we define:

$$IRL_{\mathcal{A}}(c) = 2^{(-Conf_{\mathcal{A}}(c))}$$

One can easily verify that $IRL_{\mathcal{A}}(c) \rightarrow 0$ when $Conf_{\mathcal{A}}(c) \rightarrow \infty$ and $IRL_{\mathcal{A}}(c) \rightarrow 1$ when $Conf_{\mathcal{A}}(c) \rightarrow 0$. In other words, the better the reconstruction, the higher the risk. Therefore, we can capture the informal concepts of *insecure* algorithm ($IRL_{\mathcal{A}} = 1$) and *secure* ($IRL_{\mathcal{A}} = 0$) as well.

2.2 Confidential Density Estimation

Density-based clustering is a popular technique, which reduces the search for clusters to the search for dense regions. This is accomplished by estimating a so-called probability density function from which the given data set is assumed to have arisen. An important family of method is known as *kernel estimator* [8]. Let $D = \{\mathbf{x}_i \mid i = 1, \dots, N\} \subset \mathbb{R}^n$ represent a set of data objects. Let K be a real-valuated, non-negative, non-increasing function on \mathbb{R} with finite integral over \mathbb{R} . A *kernel-based density estimate* $\hat{\varphi}_{K,h}[S](\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}_+$ is defined as follows:

$$\hat{\varphi}_{K,h}[D](\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N K\left(\frac{d(\mathbf{x}, \mathbf{x}_i)}{h}\right) \tag{1}$$

In [8] is presented an algorithm to find center-defined clusters using a density estimate. In [10] is presented the KDEEC schema, a density-based algorithm for DDC. In density-based DDC each peer contributes to the mining task with a local density estimate of the local data set and not with data (neither original nor randomized). As shown in [4], in some cases, knowing the inverse of kernel function implies in the reconstruction of original (confidential) data. Therefore, we look for a more confidential way to build the density estimate, i.e. one which doesn't allow reconstruction of data.

Definition 6. Let $f : \mathbb{R}_+ \cup \{0\} \rightarrow \mathbb{R}_+$ be a decreasing function. Let $\tau \in \mathbb{R}$ be a sampling rate and let $z \in \mathbb{Z}_+$ be an index. Denote by $\mathbf{v} \in \mathbb{R}^n$ a vector of iso-levels¹ of f , whose each component $v^{(i)}$, $i = 1, \dots, n$, is built as follow:

$$v^{(i)} = f(z \cdot \tau), \text{ if } f(z \cdot \tau) < f([z - 1] \cdot \tau)$$

Moreover $0 < v^{(0)} < v^{(1)} \dots < v^{(n)}$.

¹ One can understand \mathbf{v} as iso-lines used to contour plots.

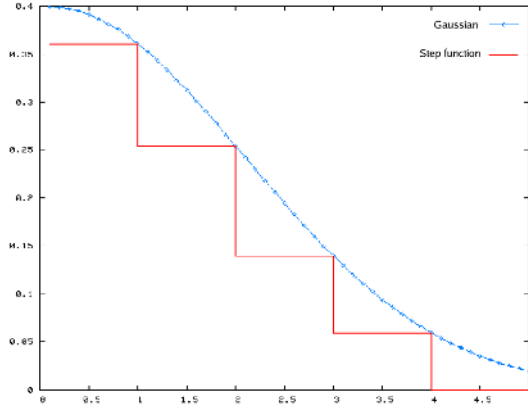


Fig. 1. $\psi_{f,\mathbf{v}}$ of the Gaussian function

Definition 7. Let $f : \mathbb{R}_+ \cup \{0\} \rightarrow \mathbb{R}$ be a decreasing function. Let \mathbf{v} be a vector of iso-levels of f . Then we define the function $\psi_{f,\mathbf{v}}$ as:

$$\psi_{f,\mathbf{v}}(x) = \begin{cases} 0, & \text{if } f(x) < v^{(0)} \\ v^{(i)}, & \text{if } v^{(i)} \leq f(x) < v^{(i+1)} \\ v^{(n)}, & \text{if } v^{(n)} \leq f(x) \end{cases} \quad (2)$$

Definitions 6 and 7 together define a step function based on the shape of some given function f . Figure 1 shows an example of $\psi_{f,\mathbf{v}}$ applied to a Gaussian² function with $\mu = 0$ and $\sigma = 2$, using four iso-levels.

Lemma 1. Let $\tau \in \mathbb{R}$ denote a sampling rate, and $z \in \mathbb{Z}_+$ be an index. Define $f_1 : \mathbb{R}_+ \rightarrow \mathbb{R}_+$, a decreasing function and \mathbf{v} , a vector of iso-levels. If we define a function $f_2 = f_1(x - k)$, then $\forall k \in (0, \tau), \forall z \in \mathbb{Z}_+$ we will have $\psi_{f_2,\mathbf{v}}(z \cdot \tau) = \psi_{f_1,\mathbf{v}}(z \cdot \tau)$.

Proof. For $k = 0$ we get $f_2(x) = f_1(x - 0)$ and its is trivial to see that the assertion holds. For $0 < k < \tau$ we have $f_2 = f_1(x - k)$. Without loss of generality, let $z > 0$ be some integer. So, $f_2(z \cdot \tau) = f_1(z \cdot \tau - k) = f_1([z - k/\tau] \cdot \tau)$. If $f_1([z - 1] \cdot \tau) = a > f_1(z \cdot \tau) = b$ then we have $\psi_{f_1,\mathbf{v}}(z \cdot \tau) = a$. Since $z - 1 < z - k/\tau < z$, and since f_1 is decreasing, $f_1([z - 1] \cdot \tau) = a > f_1([z - k/\tau] \cdot \tau) > b = f_1(z \cdot \tau)$. By the definition 7 we can write $\psi_{f_1,\mathbf{v}}([z - k/\tau] \cdot \tau) = b = \psi_{f_1,\mathbf{v}}(z \cdot \tau)$

This lemma means that we have some ambiguity associated with the function $\psi_{f,\mathbf{v}}$, given some τ and \mathbf{v} , since two functions will issue the same values iso-levels around the points close than τ .

With this definition we return to our problem of uncertainty of local density. We will substitute a kernel K by $\psi_{K,\mathbf{v}}$, Given a sample rate τ . According with

² Gaussian function is defined by $f(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-(x-\mu)^2/2\sigma^2}$.

the lemma 1, we should expect to localize the points in a interval not smaller than $|(0, \tau)|$, i.e. the confidentiality will be $Conf_{\mathcal{A}} \geq \tau$. So, we will compute a rough approximation of the local density estimate using:

$$\tilde{\varphi}[D^j](x) = \begin{cases} \sum_{x_i \in N_x} \psi_{K, \mathbf{v}}\left(\frac{d(x, x_i)}{h}\right) & , \text{if } (x \bmod \tau) = 0 \\ 0 & , \text{otherwise.} \end{cases} \tag{3}$$

where N_x denotes the neighborhood of x .

Since $\psi_{K, \mathbf{v}}$ is a non-increasing function, we can use it as a kernel function. The global approximation can be computed by: $\tilde{\varphi}[D](x) = \sum_{j=1}^p \tilde{\varphi}[D^j](x)$

3 The KDECS Algorithm

KDECS is an extension of the KDECS Schema, which is a recent approach for kernel-based distributed clustering [10]. In KDECS each site transmits the local density estimate to a helper site, which builds a global density estimate and sends it back to the peers. Using the global density estimate the sites can execute locally a density-based clustering algorithm. KDECS works in a similar way, but replaces the original estimation by an approximated value. The aim is to preserve data confidentiality while maintaining enough information to guide the clustering process.

3.1 Basic Definitions

Definition 8. Given two vectors $z_{low}, z_{high} \in \mathbb{Z}^n$, which differ in all coordinates (called the sampling corners), we define a grid G as the filled-in cube in \mathbb{Z}^n defined by z_{low}, z_{high} . Moreover for all $z \in G$, define $n_z \in \mathbb{N}$ as a unique index for z (the index code of z). Assume that z_{low} has index code zero.

Definition 9. Let G be a grid defined with some $\tau \in \mathbb{R}^n$. We define a sampling S^j of $\tilde{\varphi}[D^j]$ given a grid G , as:

$$S^j = \{\tilde{\varphi}_z^j \mid \forall z \in G, \tilde{\varphi}_z^j > 0\}$$

where $\tilde{\varphi}_z^j = \tilde{\varphi}[D^j](z \cdot \tau)$. Similarly, the global sampling set will be defined as: $S = \{\tilde{\varphi}_z \mid \forall z \in G, \tilde{\varphi}_z > 0\}$

Definition 10 (Cluster-guide). A cluster guide $CG_{i, \theta}$ is a set of index codes representing the grid points forming a region with density above some threshold θ :

$$CG_{i, \theta} = \{n_z \mid \tilde{\varphi}_z \geq \theta\}$$

such that $\forall n_{z_1}, n_{z_2} \in CG_{i, \theta} : z_1$ and z_2 are grid neighbors and $\bigcap_{i=1}^C CG_{i, \theta} = \emptyset$. A complete cluster-guide is defined by: $CG_{\theta} = \{CG_{i, \theta} \mid i = 1, \dots, C\}$ where C is the number of clusters found using a given θ .

A cluster-guide $CG_{i, \theta}$ can be viewed as a contour defining the cluster shape at level θ (a iso-line), but in fact it shows only the internal grid points and not the true border of the cluster, which should be determined using the local data set.

3.2 Detailed Description

Our algorithm has two parts: Local Peer and Helper. The local peer part of our distributed algorithm is density-based, since this was shown to be a more general approach to clustering [8].

Algorithm 1 Local Peer

Input: D^j (local data set), \mathcal{L} (list of peers), \mathcal{H} (Helper);
Output: *clusterMap*;

```

1: negotiate( $\mathcal{L}, K, h, G, \theta$ );
2:  $lde \leftarrow \text{estimate}(K, h, D^j, G, \delta)$ ;
3:  $S^j \leftarrow \text{buildSamplingSets}(lde, G, \theta, v)$ ;
4: send( $\mathcal{H}, S^j$ );
5:  $CG_\theta \leftarrow \text{request}(\mathcal{H}, \theta)$ ;
6:  $clusterMap \leftarrow \text{cluster}(CG_\theta, D^j, G)$ ;
7: return clusterMap

8: function cluster( $CG_\theta, D^j, G$ )
9:   for each  $\mathbf{x} \in D^j$  do
10:     $z \leftarrow \text{nearestGridPoint}(\mathbf{x}, G)$ ;
11:    if  $n_z \in CG_{i,\theta}$  then
12:       $clusterMap(\mathbf{x}) \leftarrow i$ ;
13:    end if
14:  end for
15:  return clusterMap;
16: end function

```

Local Peer. The first step is the function `negotiate()`, which succeeds only if an agreement on the parameters is reached. Note that the helper doesn't take part on this phase. In the second step each local peer compute its local density estimate $\tilde{\varphi}[D^j](z \cdot \tau)$ for each $z \cdot \tau$, with $z \in G$. Using the definition 9 each local peer builds its local sampling set and sends it to the helper. The clustering step (line 6 in algorithm 1) is performed as a lookup in the cluster-guide CG_θ . The function `cluster()` shows the details of the clustering step. The data object $\mathbf{x} \in D^j$ will be assigned to the cluster i , the cluster label of the nearest grid point z , if $n_z \in CG_{i,\theta}$.

Helper. Given a θ , the helper sums up all samples sets and uses definition 10 to construct the cluster-guides CG_θ . Function `buildClusterGuides()` in algorithm 2 shows the details of this step.

3.3 Performance Analysis

Time. At the local site our algorithm has time complexity $O(|G|M^j + \log(C)|D^j|)$, where $|G|$ is the grid size, M^j is the average size of the neighborhood, C is the

Algorithm 2 Helper

```

1:  $S^j \leftarrow \text{receive}(\mathcal{L});$ 
2:  $\hat{\varphi}_z[D^j] = \text{recover}(S^j);$ 
3:  $\hat{\varphi}_z \leftarrow \sum \hat{\varphi}_z[D^j];$ 
4:  $CG_\theta \leftarrow \text{buildClusterGuides}(\hat{\varphi}_z, \theta);$ 
5:  $\text{send}(\mathcal{L}, CG_\theta);$ 

6: function  $\text{buildClusterGuides}(\hat{\varphi}_z, \theta)$ 
7:    $cg \leftarrow \{n_z | \hat{\varphi}_z > \theta\};$ 
8:    $n \in cg;$ 
9:    $CG_{i,\theta} \leftarrow \{n\};$ 
10:   $i \leftarrow 0;$ 
11:  for each  $n \in cg$  do
12:    if  $\exists a((a \in \text{neighbors}(n)) \wedge (a \in cg))$  then
13:       $CG_{i,\theta} \leftarrow \{n, a\} \cup CG_{i,\theta};$ 
14:    else
15:       $i++;$ 
16:       $CG_{i,\theta} \leftarrow \{n\};$ 
17:    end if
18:     $cg \leftarrow cg \setminus CG_{i,\theta};$ 
19:  end for
20:   $CG_\theta \leftarrow \{CG_{i,\theta} | i = 1, \dots, C\};$ 
21:  return  $CG_\theta$ 
22: end function

```

number of clusters and D^j is the set of points owned by peer L^j . The first lines have complexity $O(|G|M^j)$, since the algorithm compute the density for each point z in the grid G using the subset of points in D^j which are neighbors from z , with average size M^j . Line 4 has complexity determined by the size of sampling set S^j , which is a subset of G , i.e., its complexity is $O(|G|)$. Line 5 has complexity $O(C)$. The last step (line 6) has to visit each point in D^j and for each point it has to decide its label by searching the corresponding index code in one of the cluster-guides. There are C cluster guides. Assuming the look-up time for a given cluster to be $\log(C)$ we can say that $O(\log(C)|D^j|)$ is the complexity of the last step.

Time complexity at the helper (algorithm 2) is mainly determined by the size of the total sampling set. The helper will receive from p peers at most $|G|$ sampling points. The local peer has to reconstruct and sum them up (lines 2 and 3), what takes in the worst case $O(p|G|)$ steps. Thus, the process of building the cluster-guides (line 4) will take $O(|G|)$ steps in worst case.

Communication. Each site will have at most $|S^j| < |G|$ sampling points (index-codes) to send to the helper site. The helper site has at most $|G|$ index-codes to inform back to local sites, but this size would be reduced if some compression technique is used. Moreover, our algorithm uses few rounds of messages. Each site will send one message informing the local sampling S^j set to the helper and one

(or more subsequent) message(s) requesting a cluster-guide with some desired θ . The helper will send a message informing the cluster-guides on demand.

3.4 Security Analysis

We will use two scenarios to analyze the inference risk level of KDEC-S (denoted $\text{IRL}_{\text{KDEC-S}}$). First scenario we assume that the malicious peers doesn't form collusion group, i.e. $c = 1$, and the second scenario we assume that they can form collusion group, i.e., $c \geq 2$.

Lemma 2. *Let \mathcal{L} be a mining group formed by $p > 2$ peers, one of them being the helper, and $c < p$ malicious peers form a collusion group in \mathcal{L} . Let $\tau \in \mathbb{R}$ be a sampling rate. We claim that $\text{IRL}_{\text{KDEC-S}}(c) \leq 2^{-\tau}$ for all $c > 0$.*

Proof. Assume that $c = 1$, and that each peer has only its local data set and the cluster-guides he gets from the helper. The cluster-guides, which are produced by the helper, contains only code-index representing grid points where the threshold θ is reached. This is not enough to reconstruct the original global estimation. The Helper has all sampling points from all peers, but it has neither information on the kernel nor on sampling parameters. Hence, the attackers can not use the inverse of Kernel function to reconstruct the data. The best precision of reconstruction has to be based on the cluster guides. So, one attacker may use the width of the clusters in each dimension as the best reconstruction precision. This lead to $\text{Conf}_{\text{KDEC-S}}(1) = a\tau$, with $a \in \mathbb{N}$, since each cluster will have at least a points spaced by τ in each dimension. Hence, if $c = 1$ then $\text{IRL}_{\text{KDEC-S}}(c) = 2^{-a\tau} \leq 2^{-\tau}$.

Assume $c \geq 2$. Clearly, any collusion group with at least two peers, including the helper, will produce a better result than a collusion which doesn't include the helper, since the helper can send to the colluders the original sampling sets from each peer. However, each sampling set \mathcal{S}^j was formed based on the $\tilde{\varphi}[D^j]$ (cf. eq. (3)). Using lemma 1 we expect to have $\text{Conf}_{\text{KDEC-S}}(c) = \tau$. With more colluders, say $c = 3$, one of them being the helper, there are no new information which could improve the reconstruction. Therefore, $\text{IRL}_{\text{KDEC-S}}(c) \leq 2^{-\tau}$, for all $c > 0$.

3.5 Comparison with KDEC

KDEC Scheme exploit statistical density estimation and information sampling to minimize communications cost among sites. Some of possibilities of inference attacks in KDEC were shown in[4]. Here we analyze it using our definition of inference risk.

Lemma 3. *Let $\tau \in \mathbb{R}$ be a sampling rate. Then $\text{IRL}_{\text{KDEC}}(c) > 2^{-\tau}$, for all $c > 0$.*

Proof. Since KDEC uses $y = \hat{\varphi}(\mathbf{x})$ it can be used by a malicious peer inside the group to compute the distance $d = K^{-1}(y)h$, and consequently the true

\mathbf{x}^* with $\mathbf{x}^* = \mathbf{x} + d$. Errors in this method can arise due machine precision, but they are still much smaller than τ , which in KDEEC is suggested to be $h/2$. Actually, this error is expected to be very small, since it is caused by machine precision. We remark that these results can be reached by one malicious peer alone, i.e. $Conf_{KDEC}(1) \ll \tau$. With collusion group this reconstruction may be more accurate. Therefore, $Conf_{KDEC}(c) \ll \tau$ for $c > 0$. Hence, $IRL_{KDEC}(c) > 2^{-\tau}$, for all $c > 0$.

Theorem 1. *Let $\tau \in \mathbb{R}$ be a sampling rate parameter. Using the same τ we have $IRL_{KDEC-S}(c) < IRL_{KDEC}(c)$, for all $c > 0$.*

Proof. Using lemmas 2 and 3 we verify that the assertion holds.

4 Related Work

The question of how to protect confidential information from unauthorized disclosure has stimulated much research in the data base community. This problem, known as *the inference problem*, was first studied in statistical databases and secure multi-level databases and more recently in data mining [7].

Other works in privacy preserving data mining uses secure multi-party computation (SMC) [11, 12, 9], *sanitization* [3, 5] and *data randomization* [2, 13]. Some privacy measures were proposed in [6] and [1] to the case where the mining algorithm uses randomized data. The idea of randomization seems to be promising but in a distributed set the reconstruction of local probabilities densities would lead to errors in the global density, what would lead to erroneous clustering results.

5 Conclusions

Our contribution can be summarized as: a definition of inference levels for DDM and a distributed algorithm for clustering which is inference-proof at certain level. Our definition of confidentiality and inference levels make little assumptions, what allow comparison of a broad range of data mining algorithms with respect to the risk of data reconstruction, and consequently permit us to classify them in different security classes. On the other hand, this levels are currently defined just to distributed data clustering and doesn't include (up to date) the notion of discovery of data ownership in a mining group.

KDEC-S is based on a modified way of computing density estimation such that it is not possible to reconstruct the original data with better probability than some given level. Results of our analysis using our inference risk level showed that our algorithm presents better improved security level w.r.t. inference attacks to kernel density estimate, without compromising the clustering results. One can argue that KDEEC-S has the disadvantage of using more parameters than KDEEC. However, KDEEC-S is better noise resistance than KDEEC, can find arbitrary-shape clusters (as any density-based clustering algorithm), and performs the

clustering faster, since it uses a lookup table instead of hill climbing the density estimation.

As future work we plan to apply our definition of inference level to others DDM areas.

References

1. Dakshi Agrawal and Charu C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of 20th ACM Symposium on Principles of Database Systems*, pages 247–255, Santa Barbara, California, May 2001.
2. Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proc. of the ACM SIGMOD Conference on Management of Data*, pages 439–450. ACM Press, May 2000.
3. M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. Verykios. Disclosure limitation of sensitive rules. In *Proceedings of 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99)*, pages 45–52, Chicago, IL, November 1999.
4. Josenildo C. da Silva, Matthias Klusch, Stefano Lodi, and Gianluca Moro. Inference attacks in peer-to-peer homogeneous distributed data mining. In *16th European Conference on Artificial Intelligence (ECAI 04)*, Valencia, Spain, August 2004.
5. Elena Dasseni, Vassilios S. Verykios, Ahmed K. Elmagarmid, and Elisa Bertino. Hiding association rules by using confidence and support. *Lecture Notes in Computer Science*, 2137:369–??, 2001.
6. A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *In Proceedings of PODS 03.*, San Diego, California, June 9-12 2003.
7. Csilla Farkas and Sushil Jajodia. The inference problem: A survey. *ACM SIGKDD Explorations Newsletter*, 4(2):6–11, 2002.
8. Alexander Hinneburg and Daniel A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Knowledge Discovery and Data Mining*, pages 58–65, 1998.
9. Murat Kantarcioglu and Chris Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. In *The ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'02)*, June 2002.
10. Matthias Klusch, Stefano Lodi, and Gianluca Moro. Agent-based distributed data mining: the KDEC scheme. In Matthias Klusch, Sonia Bergamaschi, Pete Edwards, and Paolo Petta, editors, *Intelligent Information Agents: the AgentLink perspective*, volume 2586 of *Lecture Notes in Computer Science*. Springer, 2003.
11. Yehuda Lindell and Benny Pinkas. Privacy preserving data mining. *Lecture Notes in Computer Science*, 1880:36–54, 2000.
12. Benny Pinkas. Cryptographic techniques for privacy-preserving data mining. *ACM SIGKDD Explorations Newsletter*, 4(2):12–19, 2002.
13. Shariq J. Rizvi and Jayant R. Haritsa. Maintaining data privacy in association rule mining. In *Proceedings of the 28th VLDB – Very Large Data Base Conference*, pages 682–693, Hong Kong, China, 2002.

A Novel Approach of Multilevel Positive and Negative Association Rule Mining for Spatial Databases

L.K. Sharma¹, O.P. Vyas¹, U.S. Tiwary², and R. Vyas¹

¹ School of Studies in Computer Science,
Pt. Ravishankar Shukla University Raipur (C.G.) 492010-India
{lksharmain, dropvyas, ranjanavyas}@gmail.com

² Indian Institute of Information Technology, Allahabad- India
ust@iiita.ac.in

Abstract. Spatial data mining is a demanding field since huge amounts of spatial data have been collected in various applications, ranging from Remote Sensing to GIS, Computer Cartography, Environmental Assessment and Planning. Although there have been efforts for spatial association rule mining, but mostly researchers discuss only the positive spatial association rules; they have not considered the spatial negative association rules. Negative association rules are very useful in some spatial problems and are capable of extracting some useful and previously unknown hidden information. We have proposed a novel approach of mining spatial positive and negative association rules. The approach applies multiple level spatial mining methods to extract interesting patterns in spatial and/or non-spatial predicates. Data and spatial predicates/association-ship are organized as set hierarchies to mine them level-by-level as required for multilevel spatial positive and negative association rules. A pruning strategy is used in our approach to efficiently reduce the search space. Further efficiency is gained by interestingness measure.

1 Introduction

A spatial association rule describes the implication of a feature or a set of features by another set of features in spatial databases. A spatial association rule [5] is a rule of the form “ $A \rightarrow B$ ”, where A and B are sets of predicates, some of which are spatial ones. In large spatial databases, many association relationships may exist but most researchers [5], [6], [7] focus only on the patterns that are relatively “Strong” i.e. the patterns that occur frequently and hold. In most cases the concepts of minimum support and minimum confidence are used. Informally the support of a pattern A in a set of spatial objects S is the probability that a member of S satisfies pattern A, and the confidence of “ $A \rightarrow B$ ”, is the probability that pattern B occurs if pattern A occurs.

A large number of robberies/crimes are committed in a large metropolitan area. A criminologist who analyzes the pattern of these robberies may visualize crime sites using a GIS system and use the maps presenting locations of other objects. A financial analyst can do the real estate investment analysis, such as price changes of houses in different localities, using maps and location-specific characteristics. It has become an

essential software tool for government services making decisions, analysis, planning and management of census, voting, mining and mineral exploration, systems for consultation and integration.

In spatial databases certain topological relationships hold at all times. Such topological relationships can be viewed as spatial association rules with 100% confidence, for example the containment relationship, which can be expressed as;

$$\text{contain}(X, Y) \wedge \text{contain}(Y, Z) \rightarrow \text{contain}(X, Z) \tag{1}$$

However a problem with such a process is that the selection of interesting patterns has to be performed only on frequent patterns. Standard association rules are not enough expressive for some applications, so we need to mine not only frequent patterns but also infrequent patterns. Mining of infrequent patterns is known to be intractable. For example in crime site analysis of a criminologist, one can analyze pattern of robberies using maps presenting locations of other objects to find patterns, but it is also very important to know the infrequent patterns involved on the location.

Unlike existing spatial mining technique, in this paper we extend the traditional spatial associations to include infrequent patterns or negative spatial association rule mining in the following form;

$$\text{contain}(X, Y) \wedge \text{contain}(Y, Z) \rightarrow \text{not_contain}(X, Z) \tag{2}$$

Mining negative spatial association rules is a difficult task due to the fact that there are essential differences between positive and negative association rule mining. In mining task, the possible negative rules can be quite more than positive association rules, but the user may not be interested in all positive and negative association rules. In this paper we also discuss how can one find out the interesting positive and negative spatial association rules. This technique makes computation faster. The rest of this paper is organized as follows. In next section we present some related concepts and definition of spatial association rule. In section 3, we discuss the pruning strategy for mining spatial positive and negative association rule. In section 4, we discuss the spatial positive and negative association rule and finally in section 5 we discuss the efficiency and other features of the algorithm and the conclusions and the scope of the future work.

2 Spatial Association Rule

Most researchers [5][6][7] used rules reflecting structure of spatial objects and spatial/spatial or spatial/nonspatial relationships that contain spatial predicates, e.g. *adjacent_to*, *near_by*, *inside*, *close_to*, *intersecting*, etc. Spatial association rules can represent object/predicate relationships containing spatial predicates. For example, the following rules are spatial association rules.

- Nonspatial consequent with spatial antecedent(s)
 $is_a(X, town) \wedge intersects(X, highway) \rightarrow adjacent_to(X, water) \dots (80\%).$
- Spatial consequent with non-spatial/spatial antecedent(s)
 $is_a(X, gas_station) \rightarrow close_to(X, highway) \dots (75\%).$

Various kinds of spatial predicates can be involved in spatial association rules. They may represent topological relationships between spatial object, such as *disjoint*, *intersects*, *inside/outside*, *adjacent_to*, *covers/covered_by*, *equal*, etc. They may also represent spatial orientation or ordering, such as *left*, *right*, *north*, *east*, etc, or contain some distance information, such as *close_to*, *far_away*, etc. For systematic study of the mining of spatial association rules, some preliminary concepts are discussed in [6], as follows;

Definition 1. A Spatial association rule is a rule of the form;

$$P_1 \wedge P_2 \wedge P_3 \wedge \dots \wedge P_m \rightarrow Q_1 \wedge Q_2 \wedge Q_3 \wedge \dots \wedge Q_n \quad (c\%) \dots \dots \dots (3)$$

Where at least one of the predicates $P_1 \wedge \dots \wedge P_m$, $Q_1 \wedge \dots \wedge Q_n$ is a spatial predicate, and $c\%$ is the confidence of the rule which indicates that $c\%$ of objects satisfying the antecedent of the rule will also satisfy the consequent of the rule.

Definition 2. A rule “ $P \rightarrow Q/S$ ” is strong if predicate “ $P \wedge Q$ ” is large in set S and confidence of “ $P \rightarrow Q/S$ ” is high.

The above definition for an association rule $P \rightarrow Q$ has a measure of the strength called confidence (denoted as *conf*) defined as the ratio $\text{supp}(P \cup Q) / \text{supp}(P)$, where $P \cup Q$ means that both P and Q are present.

Association rule discovery seeks rules of the form $P \rightarrow Q$ with support and confidence greater than or equal to, user specified support (*ms*) and minimum confidence (*mc*) thresholds respectively. This is referred to as the support-confidence framework [1] and the rule $P \rightarrow Q$ is an interesting positive association rule. An item set that meets the user specified minimum support is called frequent item set. Accordingly an infrequent item set can be defined as an item set that does not meet the user specified minimum support. Like positive rule, a negative rule $P \rightarrow \neg Q$ also has measure of its strength, confidence, defined as the ratio $\text{supp}(P \cup \neg Q) / \text{supp}(P)$ where $\text{supp}(\neg Q)$ can be measured by $1 - \text{supp}(Q)$. This infrequent item set may be significant as illustrated by following example [9].

Example 1. Let $\text{supp}(c) = 0.6$, $\text{supp}(t) = 0.4$, $\text{supp}(t \cup c) = 0.05$ and $mc = 0.52$. The confidence of $t \rightarrow c$ is $\text{supp}(t \cup c) / \text{supp}(t) = 0.05 / 0.4 = 0.125 < mc (= 0.52)$ and $\text{supp}(t \cup c) = 0.05$ is low. This indicates that $t \cup c$ is an infrequent item set and that $t \rightarrow c$ cannot be extracted as rule in support confidence framework. However, $\text{supp}(t \cup \neg c) = \text{supp}(t) - \text{supp}(t \cup c) = 0.4 - 0.05 = 0.35$ is high and the confidence of $t \rightarrow \neg c$ is the ratio $\text{supp}(t \cup \neg c) / \text{supp}(t) = 0.35 / 0.4 = 0.875 > mc$. Therefore $t \rightarrow \neg c$ is a valid rule.

By extending the definition in [5] [6] [7] negative spatial association rule discovery is proposed to be defined as follows:

Definition 3. The support of a conjunction of predicate, $P = P_1 \wedge \dots \wedge P_m$, in a set S denoted as $\text{supp}(P/S)$, is the number of objects in S which satisfy P versus the cardinality of S. The confidence of rule $P \rightarrow \neg Q$ is the ratio of $\text{supp}(P \wedge \neg Q / S)$ versus $\text{supp}(P/S)$ i.e. the possibility that a member of S does not satisfy Q when the same member of S satisfies P. A single predicate is called 1-predicate. A conjunction of k single predicates is called a k-predicate.

Our study of spatial association relationship is confined to newly formed Chhattisgarh (C.G.) state in India whose map is presented in Figure 1 with the following database relations for organizing and representing spatial objects:

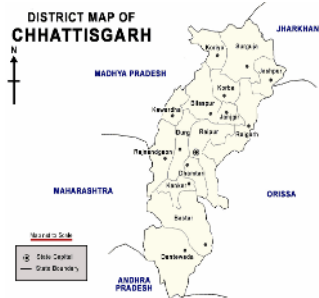


Fig. 1. Chhattishgarh State in India

- Town (town_name, town_type, population, literacy, geo...)
- Road (road_name, road_type, geo...)
- Water (water_name, water_type, geo...)
- Boundary (type, admin_region, geo...)
- Mine (mine_name, mine_type, geo...)
- Forest (forest_name, forest_type, geo...)

It may be noted that in the above relational schema, the attribute “geo” represents a spatial object (a point, line, area, etc.) whose spatial pointer is stored in a tuple of the relation and points to a geographic map. The attribute “type” of a relation is used to categorize the types of spatial objects in the relation. For example, the type for road could be {national highway, state highway, ...} and the type of water could be {rivers, lakes, ...}. The boundary could be boundary between two state regions such as Chhattisgarh and Maharastra in India.

To facilitate mining multiple level association rules and efficient processing, concept hierarchies are provided for both data and spatial predicates.

A set of hierarchies for data relations is defined as follows.

A concept hierarchy for *town*

(town (large town (big city (Raipur, Bilaspur, Durg ...)), medium size (...),...))

A concept hierarchy for *water*

(water (river (large river (Mahanadi, Kharun, Shivnath, ...)))) etc

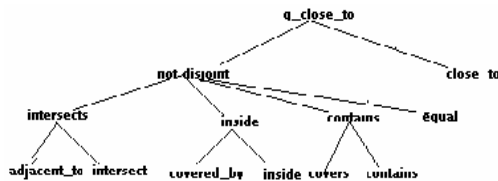


Fig. 2. Approximate spatial relations

Spatial predicates (topological relations) should also be arranged into a hierarchy for computation of approximate spatial relations (like “g_close_to see Figure - 2) using efficient algorithms with coarse resolution at a higher concept level and refining the computations when it is confined to a set of more focused candidate objects.

3 Identification of Interesting Item Set

There can be an exponential number of predicates in a database and only some of them are useful for mining association rule of interest. Therefore it is also an important issue to efficiently search the interesting itemset. In this paper we use a pruning strategy [9] to find out potentially interesting itemset. An interestingness function [9], $interest(X, Y) = |supp(X \cup Y) - supp(X) supp(Y)|$ and a threshold mi (minimum interestingness) are used. If $interest(X, Y) \geq mi$, the rule $X \rightarrow Y$ is of potential interest, an $X \cup Y$ is referred to as *potentially interesting itemset*. Using this approach, we can establish an effective pruning strategy for efficiently identifying all frequent itemsets of potential interest in a database.

Integrating this interest (X, Y) mechanism into the support-confidence framework, ‘I’ is a frequent itemset of potential interest (fipi) if:

$$\begin{aligned}
 fipi(I) &= supp(I) \geq ms \wedge \\
 &\exists X, Y: X \cup Y = I \wedge \\
 &fipis(X, Y)
 \end{aligned}
 \tag{4}$$

Where $fipis(X, Y) = X \cap Y = \emptyset \wedge$

$$f(X, Y, ms, mc, mi) = 1 \tag{5}$$

$f(X, Y, ms, mc, mi) =$

$$\frac{supp(X \cup Y) + conf(X \rightarrow Y) + interest(X, Y) - (ms + mc + mi) + 1}{|supp(X \cup Y) - ms| + |conf(X \rightarrow Y) - mc| + |interest(X, Y) - mi| + 1}$$

Where $f()$ [9] is a constraint function concerning the support, confidence and interestingness of $X \rightarrow Y$. Integrating the above insight and the interest (X, Y) mechanism into the support-confidence framework, J is an infrequent itemset of potential interest (iipis) if

$$\begin{aligned}
 iipis(J) &= supp(J) < ms \wedge \\
 &\exists X, Y: X \cup Y = J \wedge \\
 &iipis(X, Y)
 \end{aligned}
 \tag{6}$$

Where $iipis(X, Y) = X \cap Y = \emptyset \wedge$

$$g(X, \neg Y, ms, mc, mi) = 2 \tag{7}$$

$g(X, \neg Y, ms, mc, mi) = f(X, Y, ms, mc, mi) +$

$$\frac{supp(X) + supp(Y) - 2mc + 1}{|supp(X) - ms| + |supp(Y) - ms| + 1}$$

Where $g()$ [9] is a constraint function concerning $f()$ [9] and the support, confidence, and interestingness of $X \rightarrow Y$.

4 A Method for Mining Spatial Association Rules

4.1 An Example of Mining Spatial Association Rule

Example 2. We examine how the data-mining query posed in Example 1 is processed, which illustrates the method for mining spatial association rules for Chhattisgarh state.

Firstly, a set of relevant data is retrieved by execution of the data retrieval methods [2] on the data-mining query. This extracts the following data sets whose spatial portion is inside Chhattisgarh: (1) towns: only *tahsil place* (2) road: National and state highway (3) water: only rivers, lake.

Secondly the generalized *close_to* relationship between towns and the other four classes of entities is computed at a relatively coarse resolution level using a less expensive spatial algorithm such as the MBR data.

Table 1. Large k-predicate sets at the first level (for 50 towns in Chhattisgarh)

K	Large k - predicate set	Count
1	<Adjacent to, water >	29
1	<Intersect to, highway>	25
1	<close to, highway>	30
1	<close to, state boundary>	25
2	<Adjacent to, water ><Intersect to, highway>	20
2	<Adjacent to, water ><close to, highway>	20
2	<Adjacent to, water ><close to, state boundary>	18
2	<close to, highway><close to, state boundary>	15
3	<Adjacent to, water ><Intersect to, highway><close to, state boundary>	10
3	<Adjacent to, water ><close to, highway><close to, state boundary>	8

Spatial association rules can be extracted directly from table 1. To illustrate this, the object set of interest; $f(X, Y, ms, mc, mi)$ can be replaced with the following

$$f(X, Y, ms, mi) = \frac{supp(X \cup Y) + interest(X, Y) - (ms + mi) + 1}{|supp(X \cup Y) - ms| + |interest(X, Y) - mi| + 1}$$

For example (intersect highway) has support 0.5 and (adjacent to, water), (intersect, highway) has support 0.4 and if we consider $ms = 0.2$ and $mi = 0.1$ then $f(\langle \text{intersect, highway} \rangle, \langle \text{adjacent_to, water} \rangle, 0.2, 0.1) = \frac{0.4 + 0.05 - 0.1 + 1}{|0.4 - 0.2| + |0.05 - 0.1| + 1} > 1$

Table 2. Large k-predicate sets at the second level (for 50 towns in Chhattisgarh)

K	Large k - predicate set	Count
1	<Adjacent to, river>	20
1	<Intersect to, national highway>	15
1	<close to, national highway>	25
1	<close to, MP state boundary>	20
2	<Adjacent to, river ><Intersect to, National highway>	15
2	<Adjacent to, river ><close to, National highway>	15
2	<Adjacent to, river ><close to, MP boundary>	13
2	<close to, National highway><close to, MP state boundary>	10
3	<Adjacent to, river ><Intersect to, national highway><close to, MP boundary>	5
3	<Adjacent to, river ><close to, National highway><close to, MP state boundary>	5

Spatial association rules can be extracted directly from table 2. For example (close to, MP boundary) has support 0.4, (adjacent to, river) has support 0.4, and (adjacent to, river) (close to, MP boundary) has support 0.26 and if we consider $ms = 0.15$ and $mi = 0.1$ then $f(\langle \text{adjacent to, river} \rangle, \langle \text{close to, MP boundary} \rangle, 0.15, 0.1) =$

$$\frac{0.26 + 0.1 - 0.05 + 1}{|0.26 - 0.15| + |0.1 - 0.1| + 1} > 1$$

4.2 An Algorithm for Mining Spatial Association Rules

Algorithm 4.1. Mining the spatial positive and negative association rules in a large spatial database.

Input: The input consists of a spatial database, a mining query and a set of thresholds as follows:

- i. A spatial database SDB and set of concept hierarchies.
- ii. A query of a reference class set of task relevant classes for spatial object and a set of task relevant spatial relations.
- iii. Three thresholds: minimum support, minimum confidence, and minimum interestingness.

Output: Strong spatial positive and negative association rules for the relevant sets of objects and relations.

The above algorithm can be summarized in the following way:

```

Producer find_large_interested_predicate (SDB)
(1) for(l = 0 ; L[l,1] != 0 and l < max_level; l++)
(2) PL[l] ← 0; NL[l] ← 0
(3) let L[l,1]= get_predicates(SDB, l);
    PL[l] ← PL ∪ L[l,1]
(4) for (k = 2; L[l, k-1] !=0; k++)do begin
(4.1) let Pk = get_candidate_set (L [l, , k-1]);
(4.2) for each object s in S do begin
(4.3) Ps= get_subsets (P, s);//Candidate satisfied by s
(4.5) for each object set p ∈ Ps do p.supp++;
(4.6) end;
(4.7) Let L[l,k]←{p|p∈Pk∧(supp(c)=(c.supp/|SDB|)>=ms)};
(4.8) Let N [l, k] ← Pk- L [l, k];
(4.9) for each object set I in L [l, k] do
        if Not(fipi(I)) then
            let L[l,k] = L[l,k] - {I};
            let PL[l] = PL[l] ∪ L[l,k];
(4.10) for each object set J in N [l, k] do
            if NOT(iipi(J)) then
                let N[l,k] = n[l,k] - {J};
                let NL[l] = NL[l] ∪ N[l,k];
            end
        end
    end
(5) Output = generate_association_rules (PL[l], NL[l])
    end

```

In this procedure, step (1) shows that the mining of the positive and negative association rules is performed level by level, starting from the top most level until either the large 1-predicate set table is empty or it reaches the maximum concept level for each level l , step (3) computes the large 1-predicate sets and puts into table $L[l, I]$, step(4) computes the potentially frequent and infrequent itemsets, which is stored respectively as $PL[l]$, $NL[l]$ and finally the algorithm generates the spatial positive and negative association rules at each concept level from the frequent predicate table $PL[l]$ and infrequent predicate table $NL[l]$.

5 Implementation

The Algorithm explained here was implemented taking thematic map data of Chhattisgarh state of India and using programming language JAVA. The experiment was performed on a Pentium IV having 128 MB RAM.

The algorithm generated multilevel positive and negative associationships. Figure 3 shows the performance of the algorithm for generating both association rules. It is evident that the execution time is increasing with the number of objects in database but the increase for large number of positive and negative association rules is not enormous in view of the fact that the number of negative associations are

reasonably large. This justifies the use of our proposed algorithm to mine positive and negative association rules simultaneously. The algorithm proposed in this paper is efficient for mining multiple level potentially interesting spatial positive and negative association rules in spatial database. We have used a pruning strategy [9] to efficiently reduce the search space.

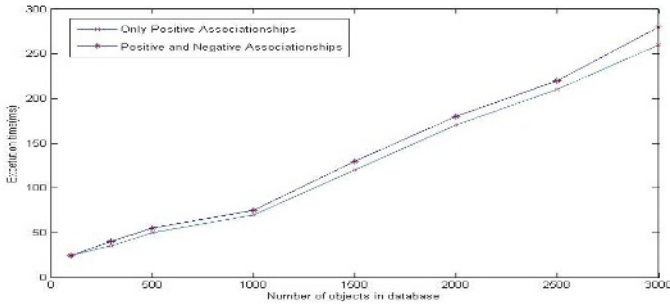


Fig. 3. Graph showing performance of ARM algorithm generating multilevel positive and negative associationships

6 Conclusion

Spatial data mining is used in areas such as remote sensing, traffic analysis, climate research, biomedical applications including medical imaging and disease diagnosis. The algorithm presented in this paper discusses efficient mining procedures for spatial positive and negative association rules. It explores techniques at multiple approximation and abstraction levels. Further, efficiency is gained by interestingness measure, which allows us to greatly reduce the number of associations needed for consideration. In our proposed approach approximate spatial computation is performed initially at an abstraction level on a large set of data, which substantially reduces the set of candidate data to be examined in the next levels. The outcome of the above mentioned spatial association rule algorithm is a set of association rules in which either the antecedent or the consequent of the rule must contain some spatial predicates (such as *close_to*):

- Non-spatial antecedent and spatial consequent: All elementary schools are located close to single-family housing developments.
- Spatial antecedent and non-spatial consequent: If a house is located in a Park, it is expensive.
- Spatial antecedent and spatial consequent: Any house that is near downtown is situated in the south of Chhattisgarh.

This algorithm works in a similar manner as the Apriori algorithm with negation and interestingness function in the “large predicate sets”. Here predicate set is a set of predicates of interest. A 1-predicate might be $\{(close_to, water)\}$, so all spatial objects that are *close_to* water will be counted as satisfying this predicate. Similarly a 2-predicate sets can be counted, and so on. In actuality the algorithm can be used to

generate multilevel positive and negative association rules at the desired coarse level or a fine level. The outcome of the algorithm can be interpreted to find the interesting associations between the spatial predicates and non-spatial predicates.

References

1. Agrawal, R., Srikant, R.: Fast algorithm for mining association rules. In Proc. 1994 Int. Conf. VLDB Santiago, Chile, Sept. (1994) 487-499.
2. Guetting, R.H.: An Introduction to Spatial Database Systems. Special Issue on Spatial Database System of the VLDB Journal, October 1994 vol 3, No 4
3. Han, J., Fu, Y.: Discovery of Multiple Level association rules from large database. Proc. of the Int. Conf. VLDB (1995) 420-431
4. Han, J., Pei, J., Yin, Y., Mao, R.: Mining Frequent Patterns with out Candidate Generation: A Frequent Pattern Tree Approach. Kluwer Publication, Netherlands (2003)
5. Malerba D. Lisi F.A.: An ILP method for spatial association rule mining. Working notes of the first workshop on Multi Relational Data mining, Freiburg, Germany (2001) 18-29.
6. Malerba D., Lisi, F. A., Analisa Appice, Francesco. : Mining Spatial Association Rules in Census Data: A Relational Approach”, 2001
7. Shekhar, S., Chawla, S., Ravadam,S., Liu, X.,Lu, C.: Spatial Databases- Accomplishments and Research Needs. IEEE Transactions on Knowledge and Data Engineering (1999) Vol. 11, No. 1
8. Smith, G.B., Bridge, S.M.: Fuzzy Spatial Data Mining. IEEE Transactions on Knowledge and Data Engineering, 2002.
9. Wu, X., Zhang, C., Zhang, S.: Efficient Mining of Both Positive and Negative Association rule. ACM Tran. On Information System (2004) Vol22.No.3, 381-405
10. Dunham, M. H.: Data Mining Introductory and Advance Topics. Pearson Education Inc, (2003)

Mixture Random Effect Model Based Meta-analysis for Medical Data Mining

Yinglong Xia*, Shifeng Weng*, Changshui Zhang**, and Shao Li

State Key Laboratory of Intelligent Technology and Systems,
Department of Automation, Tsinghua University, Beijing, China
xiayl03@mails.tsinghua.edu.cn, wengsf@tsinghua.org.cn
{zcs, shaoli}@mail.tsinghua.edu.cn

Abstract. As a powerful tool for summarizing the distributed medical information, Meta-analysis has played an important role in medical research in the past decades. In this paper, a more general statistical model for meta-analysis is proposed to integrate heterogeneous medical researches efficiently. The novel model, named mixture random effect model (MREM), is constructed by Gaussian Mixture Model (GMM) and unifies the existing fixed effect model and random effect model. The parameters of the proposed model are estimated by Markov Chain Monte Carlo (MCMC) method. Not only can MREM discover underlying structure and intrinsic heterogeneity of meta datasets, but also can imply reasonable subgroup division. These merits embody the significance of our methods for heterogeneity assessment. Both simulation results and experiments on real medical datasets demonstrate the performance of the proposed model.

1 Introduction

As the great improvement of experimental technologies, the growth of the volume of scientific data relevant to medical experiment researches is getting more and more massively. However, often the results spreading over journals and on-line database appear inconsistent or even contradict because of variance of the studies. It makes the evaluation of those studies to be difficult. Meta-analysis is statistical technique for assembling to integrate the findings of a large collection of analysis results from individual studies. Many academic papers and books have discussed the application of meta-analysis in medical researches[1].

Meta-analysis employs various statistic models to integrate available individual medical research results. Those models can be divided into fixed effect model and random effect model according to the different assumption of effect size, which is conceptualized as a standardized difference between trials for identical purpose. In the fixed effect model, the studies are assumed all to generate from a

* These two authors contribute equally to this paper.

** Corresponding author.

fixed underlying effect size; while random effect model further takes into account the extra variation[2]. The Hierarchical Bayes Linear Model (HBLM) mentioned in[3] is essentially a random effect model cooperated with prior knowledge.

Although existing meta-analysis methods have been used for decades, their intrinsic limitations lead to poor performance on complicated meta datasets. The reason is that the model assumption of those methods is too strong and therefore lack of flexibility. For example, fixed effect model regards the underlying effect size is not influenced by any other factor; while random effect model deems that the influences are centralized. In this paper, we propose a novel meta-analysis model based on Gaussian Mixture Model (GMM). The novel model, which can be viewed as an optimal linear combination of random effect models, is named *Mixture Random Effect Model* (MREM). It will be shown that traditional fixed effect methods and random effect methods are just two special cases of the proposed MREM.

2 Statistic Strategies in Meta-analysis

In meta-analysis, effect size is defined to represent the standardized performance difference between treatment group and control group in medical studies. There are multiple types of definitions of the effect size, such as risk difference (RD), relative risk (RR), odds ratio (OR), and the logarithm of the them[1]. The treatment group consists of individuals who undergo a certain medical treatment; while the control group is a collection of individuals who keep away from the treatment and just serve as reference.

Since a medical study is effected by many factors, the distribution of effect size approximates to be normal according to *Central Limit Theorem*, that is

$$y_i \sim N(\mu_i, s_i^2), \quad (1)$$

where μ_i is the underlying effect size in the i^{th} study and s_i^2 is the correspondent variance. The fixed effect model assumes the effect sizes of all studies are homogeneous and share the same fixed underlying value:

$$y_i \sim N(\mu, s_i^2), \quad (2)$$

where each study shares the same mean but different variance. Different from the fixed effect model, the random effect model considers the heterogeneity of data and assumes the distribution of underlying effect size is normal:

$$\begin{aligned} y_i &\sim N(\mu_i, s_i^2) \\ \mu_i &\sim N(\mu, \tau^2) \end{aligned} \quad (3)$$

where μ_i and s_i^2 are study-specific mean and variance respectively, that is to say, in the random effect model, μ_i is assumed to arise from a gaussian distribution with mean μ and variance τ^2 .

The Hierarchical Bayes Linear Model (HBLM) makes an improvement of random effect model. It replaces μ in Equation (3) with a linear combination of

the covariates, $x_i\beta$. Thus, it demands strong support of prior knowledge, such as correct selection and complete covariate information. Even so, HBLM is still poor for arbitrary distribution of effect size.

3 Mixture Random Effect Model

3.1 Mixture Random Effect Model

Fixed effect model and random effect model provide us two approaches to exploit the underlying structure of μ_i in meta data set. For a real meta dataset, μ_i may arise from an arbitrarily complex distribution rather than a constant in fixed effect model or a simple gaussian distribution in random effect model. Random effect model may exhibit poor performance at least in the following two cases:

1. The distribution of effect size has a unique peak but it is not a normal. Modelling μ_i as a gaussian distribution will introduce extra error.
2. The distribution of μ_i complies with a multi-peak distribution. This situation is common in real world dataset.

As mentioned, HBLM makes an effort to deal with a complex distribution by introducing covariate, $x_i\beta$. The three shortages of regression based remedy are that the covariates may not be linear additive, the selection of covariates is not easy, and the covariate information is usually unavailable for some studies in many practical cases .

Therefore, we develop a novel model to describe the characteristic of μ_i , which is expected to handle both of the two special cases listed above. Here, we propose the Mixture Random Effect Model (MREM), which is given by:

$$\begin{aligned} y_i &\sim N(\mu_i, s_i^2) \\ \mu_i &\sim \sum_{l=1}^M \alpha_l N(\xi_l, \sigma_l^2), \quad \sum_{l=1}^M \alpha_l = 1, \alpha_i > 0 \end{aligned} \quad (4)$$

Mathematically speaking, the above model utilize a gaussian mixture distribution to describe μ_i , the mean of effect size. When the number of gaussian components, M , is equal to 1, MREM degenerates to the traditional random effect model. Since μ_i in mixture random effect model is learnt unsupervisedly, we do not need any special covariate information from literatures.

After presenting MREM in this section, the problems such as choosing a proper method for parameter estimating, finding an explanation for learnt model, and proceeding subgroup analysis when the learnt gaussian components are well clustered, will be discussed in the following sections.

3.2 Parameter Estimation by Gibbs Sampling

In this section, we explore the methods to learn the parameters of the proposed MREM model from a meta dataset. As an important task in statistics and data mining, parameter estimation for mixture distribution has been explored for many years. Among these parameter estimation schemes, Expectation

Maximization (EM) algorithm [4] and Gibbs sampling [5] are two widely used methods. For the proposed mixture random effect model, we tried EM algorithm and found that there was no close form for estimating θ_l s in each iteration, so a complicated nested iteration should be performed in each EM iteration. Thus, Gibbs sampling is considered for MREM.

Gibbs sampling scheme is one of the widely used Markov Chain Monte Carlo (MCMC) routes [5] and has been applied in multidimensional sampling problems. In those problems, the joint distribution $P(x)$ is assumed to be too difficult to draw samples directly; while conditional distributions $P(x_i|\{x_j\}_{j \neq i})$ are comparatively feasible to be sampled.

In MREM, we need to estimate parameters of M gaussian components, $\Theta = \{\alpha_l, \xi_l, \sigma_l | l = 1, \dots, M\}$. Those parameters are assumed to be independent to each other. Let all priors on mixture ratio α_l , location ξ_l and logarithm variance $\log \sigma_l$ be noninformative, that is,

$$\alpha_l \sim U(0, 1), \quad \xi_l \sim U(-\infty, \infty), \quad \log \sigma_l \sim U(0, \infty) \tag{5}$$

where the prior on ξ_l , $\log \sigma_l$ are two *improper* priors in statistics. An improper prior is not integrable until it times by a likelihood function, that is, we can obtain a proper posterior distribution from an improper prior.

For each sample, we introduce a latent variable termed component indicator, $z_i \in \{1, 2, \dots, M\}$, which means that μ_i generates from gaussian component z_i . Therefore, the joint distribution is decomposed as:

$$\begin{aligned} p(\alpha, \sigma, \xi, Y, Z) &= p(\alpha, \sigma, \xi)p(Z|\alpha, \sigma, \xi)p(Y|Z, \alpha, \sigma, \xi) \\ &= p(\alpha)p(\sigma)p(\xi)p(Z|\alpha, \sigma, \xi)p(Y|Z, \alpha, \sigma, \xi) \end{aligned} \tag{6}$$

To apply Gibbs sampler, we need to find the full conditional distribution of each parameter. Since

$$p(\alpha_l | \alpha_{\bar{l}}, \sigma, \xi, Y, Z) \propto p(\alpha)p(Z|\alpha, \xi, \sigma) \propto \prod_{i=1}^K \alpha_{z_i=l} \tag{7}$$

where $\bar{l} = \{1, 2, \dots, M\} \setminus \{l\}$ and the full conditional on α is a Dirichlet distribution,

$$p(\alpha_1, \dots, \alpha_M | \sigma, \xi, Y, Z) = Dir(n_1, n_2, \dots, n_M) \tag{8}$$

where $n_l = \sum_{i=1}^K I(z_i = l)$. From Equation (6), we choose factors containing ξ , thus we have,

$$\begin{aligned} p(\xi_l | \alpha, \sigma, \xi_{\bar{l}}, Y, Z) &\propto p(\xi)p(Y|\alpha, \xi, \sigma, Z) \\ &\propto \prod_{i=1, z_i=l}^K \exp\left(-\frac{1}{2} \frac{(y_i - \xi_l)^2}{\sigma_l^2 + s_i^2}\right) \end{aligned} \tag{9}$$

Normalizing the proportion above, we get an explicit gaussian distribution,

$$p(\xi_l|\alpha, \sigma, \xi_{\bar{l}}, Y, Z) = \mathcal{N} \left(\frac{\sum_{i=1, z_i=l}^K \frac{y_i}{\sigma_l^2 + s_i^2}}{\sum_{i=1, z_i=l}^K \frac{1}{\sigma_l^2 + s_i^2}}, \frac{1}{\sum_{i=1, z_i=l}^K \frac{1}{\sigma_l^2 + s_i^2}} \right) \tag{10}$$

Now, we derive the updating formula for variance σ . From Equation (6), we have,

$$p(\sigma_l^2|\alpha, \sigma_{\bar{l}}, \xi, Y, Z) \propto p(\sigma)p(Y|\alpha, \xi, \sigma, Z) \propto \frac{1}{\sigma_l^2} \prod_{i=1, z_i=l}^K \frac{1}{\sqrt{\sigma_l^2 + s_i^2}} \exp \left(-\frac{1}{2} \frac{(y_i - \xi_l)^2}{\sigma_l^2 + s_i^2} \right) \tag{11}$$

At last, the full conditional for z_i is calculated straight forward, that is,

$$p(z_i = l|\alpha, \sigma, \xi, Y, z_{\bar{i}}) \propto p(z_i = l|\alpha, \xi, \sigma)p(y_i|\alpha, \xi, \sigma, z_i = l) \propto \alpha_l \frac{1}{\sqrt{\sigma_l^2 + s_i^2}} \exp \left(-\frac{1}{2} \frac{(y_i - \xi_l)^2}{\sigma_l^2 + s_i^2} \right) \tag{12}$$

Note that z_i is a discrete random variable valued in $\{1, 2, \dots, M\}$, where M is the total number of components of the gaussian mixture model.

3.3 Implement of Variance Updating

Once obtaining the full conditional distributions of the parameters (Equation (8), (10), (11) and (12)), we can iteratively apply Gibbs sampler for estimation. However, the full conditional distribution on σ_l is not a standard distribution and can not sample directly. Here, we employ rejection sampling with a uniform proposal function to address this problem. Therefore, the problem turns to determining the upper and lower bounds of the proposal function.

Consider a special case in which all s_i are identical, i.e. $s_i \equiv s$ and the noninformative prior, $p(\sigma_l^2) \propto 1/\sigma_l^2$. Thus, the posterior density turns to an inverse χ^2 density function:

$$p((\sigma_l^2 + s^2)|\alpha, \sigma_{\bar{l}}, \xi, Y, Z) = Inv-\chi^2(n_l, \sum_{i=1}^{n_l} (y_i - \xi_i)^2/n_l) \tag{13}$$

The above posterior is also represented equivalently as a inverse gamma distribution. Denoting sample variance $v_l = \sum_{i=1}^{n_l} (y_i - \xi_i)^2/n_l$, we obtain confidence interval of σ_l^2 with respect to given precision range, (P_{min}, P_{max}) , that is,

$$(\sigma_{min}^2, \sigma_{max}^2) = \left(\frac{n_l v_l^2}{\mathcal{I}\chi^2(P_{max}, n_l)}, \frac{n_l v_l^2}{\mathcal{I}\chi^2(P_{min}, n_l)} \right) \tag{14}$$

where $\mathcal{I}\chi^2(P, n_l)$ is inverse χ^2 cumulative distribution function with freedom degree of n_l and value of P , e.g. 2.5% and 97.5%. Let $s = \min_i(s_i)$, we have

a confidence interval $(\sigma_{min}^{(1)}, \sigma_{max}^{(1)})$; and $s = \max_i(s_i)$, we have $(\sigma_{min}^{(2)}, \sigma_{max}^{(2)})$. Therefore, we get a new interval as $(\sigma_{min}, \sigma_{max}) = (\sigma_{min}^{(1)}, \sigma_{max}^{(2)})$, within which σ_l^2 occurs with a high probability. Then, rejection sampling is consequently applied on $(\sigma_{min}, \sigma_{max})$ to generate a sample for updating σ_l^2 .

4 Model Selection and Subgroup Division

4.1 Model Selection by BIC

The number of gaussian components, M , is the only parameter that should be preset in MREM. Essentially, determining the best value of M is a model selection problem which can be solved with some feasible model selection criteria, such as AIC[6], MDL[6] and BIC[7]. Because of its broadly application, BIC is employed in MREM:

$$BIC = \log p(D|\Theta) - \frac{1}{2}d \log(K) \quad (15)$$

where D is the data, Θ is the ML estimate of the parameters, d is the number of parameters, and K is the number of data points. BIC is quite intuitive, namely, it contains a term measuring how well the parameterized model predicts the data ($\log p(D|\Theta)$) and a term which punishes the complexity of the model ($\frac{1}{2}d \log(K)$). Thus, in our algorithm, the model with the highest BIC score is selected.

4.2 Subgroup Division

One merit of MREM proposed in this paper is that it is capable to approximate arbitrary distribution, even if the distribution is very complicated. When significant disequilibrium heterogeneity exists, it is natural to divide samples into several subgroups for further study. There are two approaches for subgroup division.

The first approach is to implement division by directly observing the distributions of the gaussian components estimated in MREM, which are all one-dimensional. This approach is often feasible when the number of the gaussian components is small, or there is enough prior knowledge.

The other subgroup division approach is required when the number of components is somewhat large, and it is lack of sufficient prior knowledge. We adopt hierarchical clustering to unsupervisedly merge adjacent components. Different from an ordinary clustering task, the clustering here is applied on gaussian components. Thus, a proper measurement of dissimilarity between two gaussian components is required. Here, we employ symmetric KL divergence[8]:

$$KL(\theta_i, \theta_j) = \int_x (p(x|\theta_i) - p(x|\theta_j)) \log \frac{p(x|\theta_i)}{p(x|\theta_j)} dx \quad (16)$$

where θ_i is the parameter of the i^{th} component. The hierarchical clustering technique is an unsupervised data analysis method, which dose not demand any

prior knowledge. The results of hierarchical clustering not only reveals the proper number of subgroups, but also indicates which components should be merged in most cases.

5 Experiments

5.1 Simulation Experiment

In this section, we design an experiment on a simulated dataset to illustrate the performance of MREM. In this experiment (Experiment 1), the underlying distribution of μ_i is a GMM with three gaussian components:

$$\mu_i \sim 0.18N(x| - 2.5, 0.7^2) + 0.45N(x| - 1, 0.7^2) + 0.36N(x|2.5, 0.6^2) \quad (17)$$

We draw 150 samples from Equation (17) as the means of effect size and denote them as $\mu_1, \mu_2, \dots, \mu_{150}$. The correspondent variance s_i^2 is generated from a uniform distribution $U(0.5, 1)$. Then, a effect size y_i is drawn from $y_i \sim N(\mu_i, s_i^2)$. The task in this experiment is to approximate the distribution of μ_i given y_i and s_i for $i = 1, 2, \dots, K$. Figure 1 shows the results.

The number of components in GMMs are set from 1 to 5 respectively (see Figure 1(a) to (e)). It is found that all the iterative processes of MREM converge rapidly. The BIC score curve (see Figure 1(f)) suggests the model with 3 components is the best one, which is consistent with that of the true model.

From the selected model in Figure 1(c), we find that the left two components locate closely and they are prone to merging together as a subgroup. Therefore, it is intuitively reasonable to divide the simulated data into two subgroups.

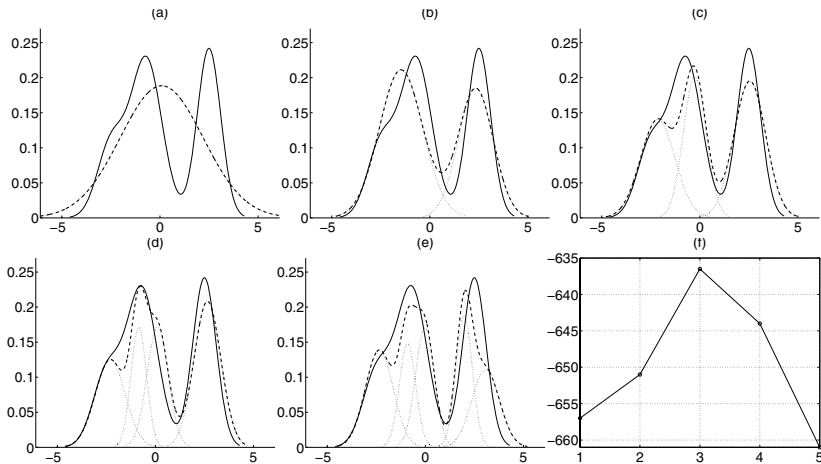


Fig. 1. Results of Experiment 1. (a)~(e) the estimation results of GMMs with 1,2,⋯,5 components respectively, where the solid curve indicates the true pdf and the dashed curve represents the estimated pdf. (f) BIC scores versus the number of components

This division is also supported by the KL divergences of the three gaussian components: $KL(\theta_1, \theta_2) = 7.86$, $KL(\theta_1, \theta_3) = 33.76$ and $KL(\theta_2, \theta_3) = 15.96$, where the three components are denoted as 1, 2 and 3 from left to right.

5.2 Real Data Experiments

In this section, we give two experiments (Experiment 2 and Experiment 3) on two real medical meta datasets. The first experiment (Experiment 2) concentrates on the level difference of the hormone factor *cortisol* in rheumatoid arthritis (RA) and healthy population. The data summarized from 15 random controlled trials [9, 10, 11, 12, 13, 14, 15] (Figure 2 (a)). The experimental results are given in Figure 2 (b) and (c).

It is found that the MREM with one gaussian component, which is equivalent to the random effect model, has the highest BIC score. The MREMs with multiple components e.g. the one shown in Figure 2(c), get smaller scores. This experiment shows that traditional random effect model is just a particular case of the proposed model. And the results illustrates the level of the hormone factor cortisol is not different between RA and healthy population. This conclusion complies with medical domain knowledge that inflammatory factor such as IL-6 is active in RA patients; while the hormone factor is not significant.

The data of the second real experiment (Experiment 3) is taken from [16] which summaries 90 randomized studies valuating the effect of *Nicotine Replacement Therapy* (NRT) on smoking cessation. Eliminating four incomplete data, we apply MREM to the remaining 86 effect sizes to find out whether the use of NRT successfully stops smoking and what its efficacy to different populations.

The result shown in Figure 3 suggests the best model is two components MREM, which presents more explicit information of underlying effect size compared to random effect model and it implies us to divide those studies into two subgroups for further study. The heterogeneity in each subgroup suggested by MREM is relatively equilibrium and their confidence intervals (CI) are listed in Table 1.

The CIs of two subgroups in Table 1 illustrate that the effect of NRT in Subgroup 2 is quite positive. While in Subgroup 1, the effect is minor, because the

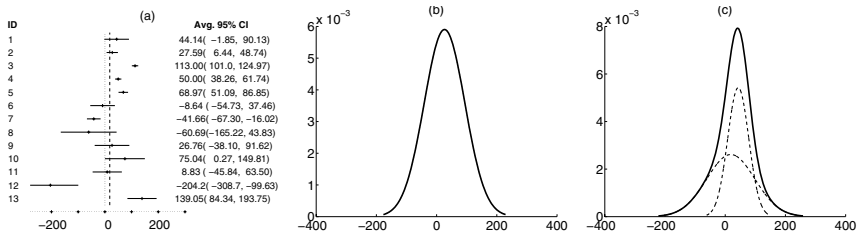


Fig. 2. Figures of Experiment 2. (a) the meta data where the short dash indicates the coordinate axes; the long dash shows the mean of data ;(b) and (c) results of random effect model and MREM respectively

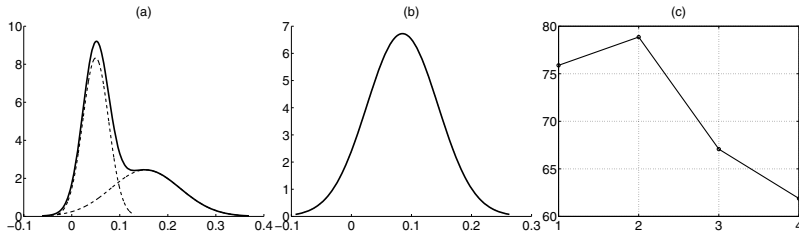


Fig. 3. Results of Experiment 3. (a) MREM with highest BIC scores. (b) estimated random effect model. (c) BIC score of models with different number of components

Table 1. The BIC scores and confidence intervals for Experiment 3

Experiment	BIC Score	95% CI
Whole data (random effect model)	75.9	(-0.0309, 0.201)
Whole data (MREM)	78.9	(-0.00874, 0.253)
Subgroup 1 (MREM)	86.2	(-0.00119, 0.0923)
Subgroup 2 (MREM)	26.8	(0.108, 0.305)

interval contains zero. This result of subgroup division obtained by MREM suggests an interesting direction for medical study. In fact, we find that Subgroup 1 mainly corresponds to women patients both in mid- and long-term follow-up and men patients in long-term follow-up; while Subgroup 2 mainly corresponds to women patients in short-term follow-up and men patients in both short- and mid-term follow-up. This division demonstrates the NRT effect is promising at short-term follow-up for all population and its efficacy becomes minor with the time lapse. Comparatively speaking, the long-term maintenance of NRT treatment gains decrease more rapidly for women than men according to our subgroup division. We find such phenomenon arising from our subgroup division consists with the medical knowledge[17][18] that NRT is efficacious both in men and women at short-term follow-up while the abstinence-rate efficacy significantly decline at long-term follow-up especially for women who suffer more from smoking cessation, such as dysphoric or depressed mood, anxiety and weight gain associated with quitting cigarettes.

6 Conclusion

In this paper, we present a novel statistical model, the mixture random effect model, for summarizing distributed heterogeneous medical studies. The proposed model unifies the traditional meta-analysis tools, that is, the fixed effect model and random effect model are just two particular cases of it. The mixture random effect model has the ability to capture arbitrary complex distribution of the effect size and provides useful information for subgroup division without prior

knowledge. We construct the model essentially by GMM, the parameters of which are estimated by MCMC approach. The novel model achieves prominent results in experiments on real clinical data, which demonstrate its potentially value for heterogeneous data analysis and medical data mining.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 30200365 and 60475001).

References

1. Whitehead, A.: *Meta-Analysis of controlled Clinical Trials*. John Wiley & Sons, New York (2002)
2. Sutton, A., Abram, K., Jones, D., Sheldon, T., Song, F.: *Methods for Meta analysis in medical Research*. John Wiley & Sons, New York (2000)
3. DuMouchel, W.H., Normand, S.T.: Computer modeling strategies for meta-analysis. In Stang, D., Berry, D., eds.: *Meta-analysis in medicine and health policy*. Marcel Dekker, New York (2000) 127–178
4. Dempster, A.P., Laird, N.M., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* **39** (1977) 1–38
5. Hastings, W.K.: Monte carlo sampling methods using markov chains and their applications. *Biometrika* **57** (1970) 97–109
6. Carlin, B.P., Louis, T.A., Carlin, B.: *Bayes and Empirical Bayes Methods for Data Analysis*. Second edn. Chapman & Hall/CRC, Florida (2000)
7. Schwarz, G.: Estimating the dimension of a model. *The Annals of Statistics* **2** (1978) 461–464
8. Duda, R., Hart, P., Stork, D.: *Pattern Classification*. Second edn. John Wiley & Sons (2001)
9. Zolil, A., et al.: Acth, cortisol and prolactin in active rheumatoid arthritis. *Clinical Rheumatology* **21** (2002) 289–293
10. Rovensky, J., et al.: Cortisol elimination from plasma in premenopausal women with rheumatoid arthritis. *Ann. Rheum. Dis.* (2003) 674–676
11. Jing, L., et al.: Circadian variation of interleukin26 and cortisol in rheumatoid arthritis. *Chin. J. Rheumatol* **6** (2002) 252–254
12. Keith, S., et al.: Adrenocorticotropin, glucocorticoid, and androgen secretion in patients with new onset synovitis/rheumatoid arthritis: Relations with indices of inflammation. *Journal of Clinical Endocrinology & Metabolism* **35** (2000)
13. Dekkers, J., et al.: Experimentally challenged reactivity of the hypothalamic pituitary adrenal axis in patients with recently diagnosed rheumatoid arthritis. *J. Rheumatol* **28** (2001) 1496–504
14. Harbuz, M., et al.: Hypothalamo- pituitary- adrenal axis dysregulation in patients with rheumatoid arthritis after the dexamethasone/corticotrophin releasing factor test. *J. Endocrinol* **178** (2003) 55–56
15. Straub, R.H., et al.: Inadequately low serum levels of steroid hormones in relation to interleukin-6 and tumor necrosis factor in untreated patients with early rheumatoid arthritis and reactive arthritis. *Arthritis Rheum.* **46** (2002) 654–662

16. Cepeda-Benito, A., Reynoso, J., Erath, S.: Meta-analysis of the efficacy of nicotine replacement therapy for smoking cessation: Differences between men and women. *Journal of Consulting and Clinical Psychology* **72** (2004) 712–722
17. Perkins, K.: Smoking cessation in women: Special considerations. *CNS Drugs* **15** (2001) 391–411
18. Cepeda-Benito, A., Reig-Ferrer, A.: Smoking consequences questionnairespanish. *Psychology of Addictive Behaviors* **14** (2000) 219–230

Semantic Analysis of Association Rules via Item Response Theory

Shinichi Hamano and Masako Sato

Department of Mathematics and Information Sciences,
College of Integrated Arts and Sciences,
Osaka Prefecture University, Sakai, Osaka 599-8531, Japan
shinchan0112@aol.com, sato@mi.cias.osakafu-u.ac.jp

Abstract. This paper aims to install Latent trait on Association Rule Mining for the semantic analysis of consumer behavior patterns. We adapt Item Response Theory, a famous educational testing model, in order to derive interesting insights from rules by Latent trait. The primary contributions of this paper are fourfold. (1) Latent trait as a unified measure can measure interestingness of derived rules and specify the features of derived rules. Although the interestingness of rules is swayed by which measure could be applied, Latent trait that combines descriptive and predictive property can represent the unified interestingness of the rules. (2) Negative Association rules can be derived without domain knowledge. (3) Causal rules can be derived and analyzed by the Graded Response Theory which is extended model of Item Response Theory. (4) The features of consumer choice that is based on the concept of multinomial logit mode in Marketing Science could be extracted. Especially the effect of promotions and product prices based on Causal rules can be generated. Our framework has many important advances for accomplishing in mining and analyzing consumer behavior patterns with diversity.

1 Introduction

In the past decade, Association Rule Mining[3] has become the focus of attention. An association rule is an implication of the form $X \Rightarrow Y$, where X and Y are item-sets satisfying $X \cap Y = \phi$ and represents the consumer purchasing patterns in a transaction. In order to measure interestingness of Association rules, many useful measures have been proposed. However respective measures have their own interestingness so that we often encounter the interpretational problems. Because there is no measure to describe the unified interestingness of Association rules, each interestingness differs much from the others. Even support and confidence, the most fundamental measures, are not efficient enough to represent the unified interestingness. Additionally there exists no measure that is not only descriptive but also predictive. The prediction and description tasks have been treated as the distinct problems for mining Association rules. As just stated, measures with well-matched properties are demanded for these purposes. We address these issues by introducing Latent trait to Association rule analysis.

An implication of the form $X \Rightarrow \neg Y$ is called a Negative Association rule[2], and represents a rule that customers who buy an item-set X are not likely to buy at least one item in an item-set Y . Negative Association rules are significant for understanding consumer behavior patterns. Although Negative Association rule mining is quite useful, too many worthless Negative Association rules could be derived. To settle this issue, Indirect Association rule mining was proposed in [3] for deriving the interesting Negative Association rules effectively.

An Indirect Association rule is an implication of the form $X \Rightarrow Y$ with $Y = \{y_1, y_2\}$ where there is a negative correlation between y_1 and y_2 . In the frameworks in [3, 4] an item-set X called a mediator represents a common item-set for the consumers, i.e., X illustrates the similarity of consumer behavior patterns. The relations of X and each of items in Y illustrate that the consumers who bought all items of a mediator X take different actions of buying an item y_1 or y_2 . For example, if we discover an Indirect Association rule ($X \Rightarrow y_1, X \Rightarrow y_2$), consumers that buy all items in X tend to buy either y_1 or y_2 but not both. It enables us to figure out that the items y_1 and y_2 are in a choice item-set, i.e., they could be competitive products or alternative products.

We have previously extended the model of an Indirect Association rule in order to represent interesting consumer behavior patterns. An Indirect Association rule due to S. Hamano and M. Sato[4] is an implication of the form $X \Rightarrow \beta_{y_j}$ with $\beta_{y_j} \in \{y_j, \neg y_j\}$ for $j = 1, 2$ and illustrates behaviors of consumer choice with two alternatives. For example, if we discover an Indirect Association rule ($X \Rightarrow y_1, X \Rightarrow \neg y_2$), consumers that buy all items in X tend to buy together with y_1 but not with y_2 . Even though an Indirect Association rule is quite valuable to recognize consumer behavior patterns, it can describe only consumer choice between two alternatives. In order to illustrate consumer choice behavior with multiple alternatives, we are going to introduce a new framework of Association rule analysis via Item Response Theory[8].

IRT is a mathematical model in educational testing for studying individual responses. The aim of IRT is to obtain fine estimates of Latent trait called ability. This is an attractive model to predict the probability of each correct response as a function of the Latent trait and some parameters. Latent trait is a powerful unobserved factor of measuring interestingness of derived rules as a unified measurement and specify the features of derived rules. Latent trait is estimated by the EM algorithm[5] with some parameters. These parameters are also significant as well as Latent trait in terms of capturing the features of rules. One of the parameter a called discrimination parameter can classify Association rules and Negative Association rules without domain knowledge and remove uninteresting rules effectively. The parameter c called guessing parameter can remove trivial rules without asking experts.

Graded Response Theory which is the extended model of IRT can be applied to Causal rule mining[7]. From this application, the effect of promotions and product prices based on Association rules can be generated like multinomial logit model[6]. As you have seen, our framework has many important advances for accomplishing in mining and analyzing consumer behavior patterns with diversity.

This paper is organized as follows. In the next section, Item Response Theory for analysis of Association rules is introduced. In Section 3, we present Graded Response Theory for analysis of Causal rules. Experimental results are presented in Section 4.

2 Semantic Analysis of Association Rules via IRT

2.1 Preliminary

Let I be a finite set of items, and D be a set of customer transactions, called a *database*, where each transaction T is a set of items such that $T \subseteq I$. Each transaction has a unique identifier i , called Transaction ID (*TID*), is denoted by T_i . We denote subsets of I , called *item-sets*, by X, Y and items of X and Y by x_1, x_2, x_3, \dots and by y_1, y_2, y_3, \dots respectively. In this paper, an item-set X means not only the subset of I but also the event that a transaction contains all items in the set X , and $Pr(X)$ denotes the probability that a transaction contains the set X . Moreover, $\neg X$ denotes the negation of the event X , i.e., the event that a transaction does not contain at least one item in X , and thus $\neg\neg X = X$ and $Pr(\neg X) = 1 - Pr(X)$.

We consider an item-set Y such that $Y = \{y_1, y_2, \dots, y_n\}$. Let y_{ij} represent a presence or an absence of an item y_j in a transaction with $TID = i$, that is, $y_{ij} = 1$ if $y_j \in T_i$ and $y_{ij} = 0$ if $y_j \notin T_i$. Item response data with $TID = i$ for given an item-set Y , denoted by $T_{i,Y}$, are represented as a binary vector, i.e., $T_{i,Y} = (y_{i1}, y_{i2}, \dots, y_{in})$. Remark that $T_i \cap Y = \{y_j | y_{ij} = 1, j = 1, 2, \dots, n\}$. For example, consider an item-set $Y = \{y_1, y_2, y_3, y_4, y_5\}$. The item response data $T_{4,Y} = (1, 0, 0, 0, 1)$ indicate that the transaction with $TID = 4$ contains items y_1 and y_5 , but does not contain items y_2, y_3 , and y_4 .

2.2 Semantic Analysis of Association Rules via IRT

In this section, we introduce Item Response Theory for an Association rule with Multiple Alternatives. Let X and $Y = \{y_1, y_2, \dots, y_n\}$ be item-sets such that $X, Y \subseteq I$ and $X \cap Y = \phi$. Let D_X be the set of transactions which contains all items in X . We assume the condition that all items in Y are statistically independent in the database D_X , called the local independence condition. We denote an Association rule with Multiple Alternatives ($X \Rightarrow \beta_{y_1}, X \Rightarrow \beta_{y_2}, \dots, X \Rightarrow \beta_{y_n}$) with $\beta_{y_j} \in \{y_j, \neg y_j\}$ by $(X; \beta_{y_1}, \beta_{y_2}, \dots, \beta_{y_n})$.

We presume that there is a Latent trait between item-sets X and Y . Let θ be a Latent trait that is an unobserved factor of measuring the underlying ability and θ_X represent the ability of an transaction that contains all items in X . That is, D_x has an inherent unobserved variable, Latent trait θ_X , and this θ_X dominates the probability of occurring an item y_j . Let $p_j(\theta_X)$ and $q_j(\theta_X)$ represent the probabilities of the presence and the absence of an item $y_j \in Y$ in D_X respectively as follows:

$$p_j(\theta_X) = Pr(y_j|X), \quad q_j(\theta_X) = 1 - p_j(\theta_X) = Pr(\neg y_j|X). \tag{1}$$

The interestingness of an Association rule $X \Rightarrow y_j$ is measured by the conditional probability of y_j given X while the interestingness of a Negative Association rule $X \Rightarrow \neg y_j$ is measured by the conditional probability of $\neg y_j$ given X . The measures λ for evaluating interestingness of an Association rule and a Negative Association rule are defined as follows:

$$\lambda(X \Rightarrow y_j) = Pr(y_j|X), \quad \lambda(X \Rightarrow \neg y_j) = Pr(\neg y_j|X). \tag{2}$$

Because there is a Latent trait between item-sets X and Y and the Latent trait θ_X of D_X dominates the probability of occurring y_j , the conditional probability can be represented as a function of θ_X . Therefore $p_j(\theta_X)$ and $q_j(\theta_X)$ represent the interestingness of an Association rule and a Negative Association rule respectively. The probability of that a transaction with $TID = i$ whether contains an item y_j can be represented briefly as follows:

$$f(y_{ij}|\theta_X) = p_j(\theta_X)^{y_{ij}} q_j(\theta_X)^{1-y_{ij}}. \tag{3}$$

For item response data $T_{i,Y} = (y_{i1}, y_{i2}, \dots, y_{in})$, translating process into a joint probability model based on the local independence assumption results in the following:

$$f(T_{i,Y}|\theta_X) = \prod_{j=1}^n p_j(\theta_X)^{y_{ij}} q_j(\theta_X)^{1-y_{ij}}. \tag{4}$$

Logarithmic likelihood function of the above probability is illustrated as follows:

$$\log L(T_{i,Y}|\theta_X) = \sum_{j=1}^n [y_{ij} \log p_j(\theta_X) + (1 - y_{ij}) \log q_j(\theta_X)]. \tag{5}$$

The parameters θ_X is estimated by maximization of the above Logarithmic likelihood function.

IRT is a prominent mathematical model in educational testing for studying individual responses. The aim of IRT is to obtain fine estimates of Latent trait called ability. This is an attractive model to predict the probability of each correct response as a function of the Latent trait and some parameters. The function called an Item Response Function (IRF) of two parameter logistic model (2PL) is defined as follows:

$$p_j(\theta_X) = \frac{1}{1 + \exp(-1.7a_j(\theta_X - b_j))} \tag{2PL} \tag{6}$$

where a_j is the discrimination parameter, b_j is the difficulty parameter and θ_X is the ability level. The parameters a_j , b_j and θ_X are estimated by maximization of the above Logarithmic likelihood function. We apply the IRF defined above to Association Rule Mining for measuring interestingness of the rules.

The IRF for Association Rule Mining represents the conditional probability of y_j given X where θ_X is the Latent trait of causing the event y_j . These parameters a_j , b_j and θ_X could be generated from the database by the EM algorithm[5].

In our new perspective, Association and Negative Association rules are discriminated by the parameter a_j . If $a_j > 0$, the Association rule $X \Rightarrow y_j$ is derived because the conditional probability of y_j given X monotonically increases with θ_X . On the other hand, if $a_j < 0$, the Negative association rule $X \Rightarrow \neg y_j$ is derived because the conditional probability of y_j given X monotonically decreases with θ_X . Hence, the parameter a_j is the key factor of distinguishing Association and Negative Association rules. Moreover the higher the absolute value of the discrimination parameter is, the more interesting the rule is. The Fisher information of an item-set Y is defined as follows:

$$I(\theta_X) = E \left[\frac{\partial}{\partial \theta} \log L(T_{i,Y} | \theta_X)^2 \right] = 1.7^2 \sum_{j=1}^n a_j^2 p_j(\theta_X) q_j(\theta_X) \tag{7}$$

Moreover the Fisher information of an item y_j is defined as follows:

$$I_{y_j}(\theta_X) = 1.7^2 a_j^2 p_j(\theta_X) q_j(\theta_X). \tag{8}$$

The discrimination parameter a_j for an item y_j is noticeably significant in terms of augment the information. Therefore the parameter a_j is applied for measuring interestingness of the rules and classifying Association rules and Negative Association rules.

The parameter b_j represents how much the occurrence of X relates the occurrence of each of items in a set of alternatives, i.e., causal relationships between X and each of items could be measured.

The most important and fascinating factor is the parameter θ_X that represents the Latent trait. By estimating the parameter θ_X , the responses of each rule could be predicted. When the three parameters are reasonably accurate, the predictive property will be assured. The parameter θ_X represents the interestingness of the rules. The higher the ability level is, the more interesting the rule is. We can say that these parameters are descriptive measures for derived rules. Item Characteristic Curve (ICC) is a graph of Item Response Function and a visual tool of illustrating the choice probability of each items.

According to the Latent trait model, the local independence condition is necessary for estimation of Latent trait and some parameters. That is, the condition of that correlations of any items are all statistically independent is absolutely necessary in order to generate fine estimates. However it is expensive to generate an item-set that satisfy the local independence condition. Therefore we introduce choice item-set for relaxing the condition of local independence in the next section.

2.3 Common Item-Sets and Choice Item-Sets

In order to apply IRT to analysis of an Association rule with Multiple Alternatives, we introduce two item-sets called common item-set and choice item-set. The common item-sets help to reduce search space and the number of uninteresting and trivial rules. Common item-sets illustrate the consumer behavior patterns as ordinal patterns. The common item-set O is defined as follows:

$$O = \{X \subset I \mid Pr(X) \geq \eta_f\},$$

where η_f is a *common item-set threshold* predefined by the user. We should note that it is desirable that there are at least 500 transactions containing all items in a common item-set for stable parameter estimations of the Latent trait. Even though common item-sets can describe the similar consumer behaviors, it is not sufficient enough to recognize the characteristics of consumers. In order to illustrate characteristics, we are going to define choice item-set as a set of Multiple Alternatives. The items in choice item-set could give us great insights by deriving with a common item-set because we could see many kinds of different selecting actions of consumers as characteristics. Let \mathfrak{S} be a family of sets as follows:

$$\mathfrak{S} = \{Y \subset I \mid \forall i_j, \forall i_k \in Y, |\rho(i_j, i_k)| \leq \eta_\rho, j \neq k, |Y| \geq 2\},$$

where ρ is the coefficient of correlation between the pair of items and η_ρ is a correlation threshold predefined by the user. Let us define an choice item-set S that is a maximal set in \mathfrak{S} , i.e., there is no sets S' in \mathfrak{S} satisfies $S \subsetneq S'$. We should also note that it is desirable that the size of a choice item-set has to be secured to a certain degree for stable parameter estimations of the Latent trait.

Definition 1. Let X be a common item-set and Y be a choice item-set where $X, Y \subseteq I$ and $X \cap Y = \phi$ and a_j be a discrimination parameter of an item y_j . An Association rule with Multiple Alternatives can be extracted as an interesting rule for 2PL IRT model if

$$(1) |a_j| \geq \eta_d \quad (\text{Discrimination parameter Condition}).$$

In the next section, we introduce Graded Response Theory for analysis of Causal rule which is extension of Association rule.

3 Causal Rule Analysis via Graded Response Theory

A Causal rule proposed in [7] is an implication of the form $X \Rightarrow Y$ where X and Y are sets of categorical variables with $X \cap Y = \phi$. For each categorical variable X_i , $R(X_i)$ called range consists of finite order categorical items. We assume that X is a conjunction of explanatory categorical variables X_i and y_j is a target item. By adapting Graded Response Theory (GRT) which is extended model of IRT, interesting insights from Causal rules could be derived such as influence and interestingness of categorical variables. Let X_i be categorical variable that has K ordered value as follows:

$$X_i = 0, 1, 2, \dots, k, \dots, K - 1 \tag{9}$$

The probability of $X_i = k$ is defined as follows:

$$p(X_i = k | \theta_{y_j}) = p_{ik} = p_{ik}^*(\theta_{y_j}) - p_{ik+1}^*(\theta_{y_j}), \tag{10}$$

where $p_{ik}^*(\theta_{y_j})$ represents the probability of that $X_i \geq k$. Note that $p_{i0}^*(\theta_{y_j}) = 1$ and $p_{iK}^*(\theta_{y_j}) = 0$. The probability $p_{ik}^*(\theta_{y_j})$ for 2PL logistic model is defined as follows:

$$p_{ik}^*(\theta_{y_j}) = \frac{1}{1 + \exp(-1.7a_i(\theta_{y_j} - b_{ik}^*))}. \tag{11}$$

Let M_l be the categorical response data matrix. By the local independence condition, the probability of an observation M_l is illustrated as follows:

$$p(M_l|\theta_{y_j}) = \prod_{i=1}^n \prod_{k=0}^{K-1} p_{ik}(\theta_{y_j})^{X_{ik}}. \tag{12}$$

Logarithmic likelihood function of the above probability is defined as follows:

$$\log L(M_l|\theta_{y_j}) = \sum_{i=1}^n \sum_{k=0}^{K-1} X_{ik} \log p_{ik}(\theta_{y_j}). \tag{13}$$

The parameters a_i, b_{ik}^* are estimated by the EM algorithm as maximization of the above Logarithmic likelihood function. The Fisher information for a target item y_j is defined as follows:

$$I_{X_i}(\theta_{y_j}) = 1.7^2 a_i^2 \sum_{k=0}^{K-1} \frac{(p_{ik}^*(\theta_{y_j})q_{ik}^*(\theta_{y_j}) - p_{ik+1}^*(\theta_{y_j})q_{ik+1}^*(\theta_{y_j}))^2}{p_{ik}(\theta_{y_j})} \tag{14}$$

Let $\hat{\theta}_X$ and $\hat{\theta}_{X_i}$ be an estimated Latent trait of X and X_i respectively for a target item y_j . The estimated Latent trait $\hat{\theta}$ is regarded significant as much as a discrimination parameter. Hence an estimated Latent trait is one of criteria for measuring the interestingness of derived Causal rules.

Definition 2. Let $X = \{X_1, X_2, \dots, X_n\}$ be a set of explanatory categorical variables and y_j be a target item. Let $\hat{\theta}_X$ and $\hat{\theta}_{X_i}$ be an estimated Latent trait of a set X of explanatory categorical variables and each explanatory categorical variables X_i respectively for a target item y_j . Causal rule $X \Rightarrow y_j$ can be extracted as an interesting causal rule, if

- (1) $\hat{\theta}_X \geq \frac{1}{n} \sum_{i=1}^n \hat{\theta}_{X_i}$, (Latent Trait Condition)
- (2) $|a_i| \geq \eta_d$, (Discrimination Condition).

3.1 Analyzing Effect of Promotion and Price

Multinomial logit model[6] have been contributed Marketing Science for identifying the variables that affect consumer choices from among a set of alternatives such as price, promotion, and so on. Even though data of price and promotion are significant factors for marketing, there have been no effective method to analyze the effect of promotions and product prices based on Association rules for market basket data. Our framework can analyze them by introducing variables for price and promotion data as explanatory categorical variables for a target item. Let $Promo_{y_j}$ be a promotion variable and $Price_{y_j}$ be a price variable for a target item y_j . A Latent trait and a discrimination parameter for each explanatory variables can be estimated by the EM algorithm. The higher discrimination

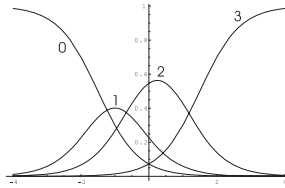


Fig. 1. Item Characteristic Curve

parameter is, the more influential and significant variable is. For example, suppose there are three explanatory variables X_1 , $Promo_{y_1}$ and $Price_{y_1}$ for a target item y_1 , i.e., $X = \{X_1, Promo_{y_1}, Price_{y_1}\}$ and the estimated discrimination parameters for each variables are 1.0, 1.5 and 0.5 respectively. As easily seen, the Promotion variable is more significant than the Price variable. Hence the promotion is the most influential factor for consumers who bought a target item y_1 . Moreover the explanatory variable X_1 is more significant than the Price variable. This could be an interesting insight because there is the factor that is more influential than the price of an item y_1 . As a result, we can recommend a marketing manager that the pricing for an item y_1 should be reconsidered. Moreover the higher Latent trait is, the more significant categorical variable is. For example, three kinds of promotions have been done for a target item y_1 . The promotion categorical variable is presented as 1, 2, and 3 for "only special display promotion", "only advertisement promotion", and "both special display and advertisement promotion" respectively. Note that the promotion variable 0 means "no promotion". If estimated Latent traits are -1.5, 0.5, 1.0, 1.5 for each promotion categorical variable 0, 1, 2, and 3 respectively, then the ICC as in figure 1 is represented. The ICC indicates that most effective promotion was "both special display and advertisement promotion" and the promotion "only special display" was not effective.

4 Experiments

We have performed analysis of Direct Marketing data distributed for the KDD CUP in 1998. These experiments are for deriving interesting Causal rules based on the graded response model and analyzing categorical variables. Suppose a Causal rule has three categorical variables for a set X of categorical variable, i.e., $X = \{X_1, X_2, X_3\}$. The domains for categorical variables X_1 , X_2 , X_3 , and Y are $X_1 = \{F, N, A, L, S\}$, $X_2 = \{1, 2, 3, 4\}$, $X_3 = \{A, B, C, D, E, F, G\}$, and $Y = \{Donor\}$. Due to the space limitation, we present only the results of RFA_2 and RFA_3 .

Figure 3 depicts the ICC of estimated parameters of each categorical variable from the data of RFA_2 and RFA_3 . The domain of the first categorical variable X_1 in RFA_2 is $\{L\}$, that is, all donors belong to the category $\{L\}$. Although the higher discrimination parameter is, the more significant the categorical variable is, there is hardly difference between the item discrimination parameters for X_2

Input : D- Database
 y_i - Target item
 t_d - minimum discrimination parameter threshold
 Output : A list of interesting Causal rules

- 1) for each explanatory categorical variable
- 2) Estimate (a_{X_i}, θ_{X_i}) by the EM algorithm
- 3) if $|a_{X_i}| \geq t_d$ then
- 4) $X = X \cup \{X_i\}$
- 5) for $X = \{X_1, X_2, \dots, X_n\}$
- 6) Estimate θ_X by the EM algorithm
- 7) if $\theta_X \geq \frac{1}{n} \sum_{k=1}^n \theta_k$ then
- 8) Output (X, y_i, θ_X)

	RFA_2	RFA_3
1	L4D	S4D
2	L4E	S4E
3	L3D	A4D
4	L3E	N4D
5	L4E	S4F

Top 5 interesting Causal rules

Algorithm of mining Causal rules via Graded Response Theory

Fig. 2. Algorithm and top 5 interesting Causal rules

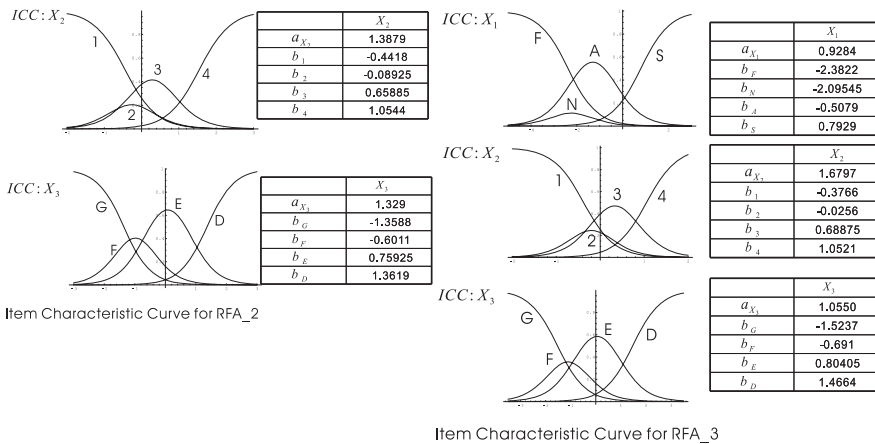


Fig. 3. Experimental results

variables. The fact that the highest item difficulty parameter of X_2 is a category 4 indicates that people who belong to category 4 are likely to be Donor. Low item difficulty parameters for categories 1 and 2 indicate that people who belong to these categories are not likely to be Donor. Although category 3 has a high difficulty parameter, it could not be an attractive category because its highest probability is not high enough. The highest item difficulty parameter of X_3 is D so that the people are potentially attractive Donor. Additionally, the second highest parameter E is attractive enough because the highest probability is high enough. The domain of the first categorical variable X_1 in RFA_3 is $\{F, N, A, S\}$, of the second variable X_2 is $\{1, 2, 3, 4\}$ and of the third variable X_3 is $\{D, E, F, G\}$. Because categories A, B and C in the third category are too low to generate secure estimates. I and L in the first category are too low as

well as A , B and C . The ICC and parameter estimations indicate S , 3, 4 and G are attractive categories for Donor. The most interesting rules are $L4D$ and $S4D$ for $RFA.2$ and $RFA.3$ respectively. Therefore the people who belong to these categories have propensity for being Donor.

5 Conclusion and Future Work

We are convinced that our framework produce good results from semantic analysis of Association rules and Causal rules. As a future work, we are going to extend the current model to relax local independence condition.

References

1. R. Agrawal, T. Imielinski and A. Swami: *Mining Association Rules between Sets of Items in Large Databases*, in Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., USA, May 26–28, 1993, pp. 207–216, 1993.
2. A. Savasere, E. Omiecinski and S. Navathe: *Mining for Strong Negative Associations in a Large Database of Customer Transactions*, in Proceedings of the Fourteenth International Conference on Data Engineering, Orlando, Florida, USA, February 23–27, 1998, pp. 494–502, 1998.
3. P.N. Tan: *Discovery of Indirect Association and Its Applications*, Dr. Thesis, UNIVERSITY OF MINNESOTA, July, 2002.
4. Shinichi Hamano and Masako Sato: *Mining Indirect Association Rules*, in Proceedings of the 4th Industrial Conference on Data Mining, ICDM2004, Leipzig, Germany, July 4-7, 2004, Lecture Notes in Computer Science 3275, pp. 106–116, 2004.
5. A. Dempster, N. Laird, and D. Rubin: *Maximum likelihood from incomplete data via the em algorithm*, Journal of Royal Statistical Society, B(39):1–38, 1977.
6. D.McFadden: *Conditional logit analysis of qualitative choice behavior*, Frontiers in Economics, pp. 105–142, 1974.
7. Shichao Zhang and Chengqi Zhang: *Discovering Causality in Large Databases*, Applied Artificial Intelligence, Vol. 16, 5(2002): 333-358.
8. Anne Boomsma, Marijtje A. J. van Duijn and Tom A. B. Snijders: *Essays on Item Response Theory* Lecture Notes in Statistics **157**, 2001.

Temporal Approach to Association Rule Mining Using T-Tree and P-Tree

Keshri Verma, O.P. Vyas, and Ranjana Vyas

School of Studies in Computer Science,
Pt. Ravishankar Shukla University Raipur Chhattisgarh – 492010, India
{seema2000_2001, vyasranjana}@yahoo.com,
opvyas@rediffmail.com

Abstract. The real transactional databases often exhibit temporal characteristic and time varying behavior. Temporal association rule has thus become an active area of research. A calendar unit such as months and days, clock units such as hours and seconds and specialized units such as business days and academic years, play a major role in a wide range of information system applications. The calendar-based pattern has already been proposed by researchers to restrict the time-based associations. This paper proposes a novel algorithm to find association rule on time dependent data using efficient T tree and P-tree data structures. The algorithm elaborates the significant advantage in terms of time and memory while incorporating time dimension. Our approach of scanning based on time-intervals yields smaller dataset for a given valid interval thus reducing the processing time. This approach is implemented on a synthetic dataset and result shows that temporal TFP tree gives better performance over a TFP tree approach.

1 Introduction

Efficient algorithm for finding frequent patterns has been one of the key success stories of data mining. The Apriori algorithm [1] by Agrawal et al is based on support-confidence framework. It begins with counting the support of each and every item of transaction table. It comprises of two phases; (1) Candidate generation in all possible combinations. (2) The database scanning and counting of all transactions for each itemset. The process continues as long as frequent item is available. There are some pertinent questions which can be raised like : What about mining the -

- Associationship in all patterns of a certain type during a specific time interval.
- Associationship in all patterns of a certain type with a specific periodicity.
- Associationship in all patterns of certain type with a specific periodicity during a specific time interval.

All these statements indicate that data is dependent on time. Time plays an important role in real dataset. In various business setups such as stock market and share market time is most important dimension In existing algorithms like Apriori [1]

and Fp-growth[7] when time aspect is involved in the dataset, it provides useful information for business, but it increases the time complexity because it is needful to scan the database for every valid specific interval. Though Fp-growth is one of the well-known approaches for finding frequent itemset, but it performs miserably when the nature of data is sparse. Sparse data creates large number of nodes and if temporal dimension is also incorporated on dataset, it again increases the branches of tree and is thus difficult to fit in memory. In such case Apriori outperforms Fp-growth approach.

Our proposed approach is aimed at devising an efficient algorithm for mining association rule in time-based dataset by using efficient data storage mechanism-Temporal T-tree.

Definition 1: A temporal pattern is a triplet $\langle \text{pattern}, \text{periodicexp}, \text{interval exp} \rangle$, where pattern is a general pattern which may be a trend, a classification rule, an association a causal relationship etc. Periodicexp is a periodic time expression or a special symbol p_null with $\phi(p_null)$ being $\{T\}$ and interval exp is a general interval expression or a special symbol I_null with $\phi(p_null)$ being $\{T\}$. It expresses that pattern hold during each interval in $\phi(\text{periodicexp} \cap \text{interval exp})$. T is the time domain. [10]

Given a time-stamped dataset D over a time domain T, the problem of mining temporal pattern of a certain type is to discover all pattern of the form $\langle \text{pattern}, \text{periodicexp}, \text{interval exp} \rangle$ in D which satisfy all the user defined threshold with respect to described minimum frequency $\text{min}_f\%$ with some given condition.

Three properties of the algorithm are of interest when considering its performance:

- The number of database access required
- The no. of computational step required in counting subset of records
- The memory requirements.

For small values of n these set of algorithm are appropriate, but large dataset these algorithms are computationally infeasible.

The rest of the paper is organized as follows : Section 2, discusses some related works. In section 3 defines temporal association rule in term of calendar schema. In Section 4 elaborate the proposed work, section 5 shows the experimental study and section 6 elaborates conclusion and future works and section 6 provides application of above investigation.

2 Related Work

The concept of association rule was introduced as Apriori algorithm [1]. Its performance was improved by deploying frequent-pattern growth approach [7]. In paper [6] the omission of the time dimension in association rule was very clearly mentioned. A temporal aspect of association rule was given by Juan [5]. According to this the transactions in the database are time stamped and time interval is specified by the user to divide the data into disjoint segments, like month, days and years. Further The cyclic association rule was introduced by Ozden [6] with minimum support and high confidence. Using the definition of cyclic association rule, It may not have high support and confidence for the entire transactional database. A nice bibliography of

temporal data mining can be found in the Roddick literature[8]. Rainsford and Roddick presented extension to association rules to accommodate temporal semantics. According to [9] logic the technique first searches the associationship than it is used to incorporate temporal semantics. It can be used in point based and interval based model of time simultaneously[9]. A Frequent pattern approach for mining the time sensitive data was introduced in[4] where the pattern frequency history under a tilted-time window framework is used to answer time-sensitive queries. A collection of item patterns along with their frequency histories are compressed and stored using a tree structure similar to FP-tree are updated incrementally with incoming transactions [4]. Li et. al. addresses the calendar based association rule problem [11],the result shows temporal apriori is 5 to 22 times faster than direct apriori, for fuzzy match temporal apriori is 2.5 to 12 times faster than direct apriori and the execution time extremely decreases with respect to precise match or fuzzy match.

3 Problem Definition

3.1 Association Rule

The concept of association rule, which was motivated by market basket analysis and was originally presented by Agrawal. [1]. Given a set of T of transaction, an association rule of the form $X \rightarrow Y$ is a relationship between the two disjoint itemsets X and Y . An association rule satisfies some user-given requirements. The support of an itemset by the set of transaction is the fraction of transaction that contain the itemset. An itemset is said to be large if its support exceeds a user-given threshold minimum support. The confidence $X \rightarrow Y$ over T is a transaction containing X and also containing Y . Due to complex candidate generation in the data set Jiewai Han invented a new technique of FP-growth method for mining frequent pattern without candidate generation [7]. In our opinion this mining associationship will become more useful if we include the time factor in to it.

3.2 Temporal Association Rule

Definition 2: The frequency of an itemset over a time period T is the number of transactions in which it occurs divided by total number of transaction over a time period. In the same way , confidence of a item with another item is the transaction of both items over the period divided by first item of that period.

Support(A) = Frequency of occurrences of A in specified time interval / Total no of Tuples in specified time interval

Confidence(A => B[Ts,Te]) = Support_count(A U B) over Interval / occurrence of A in interval

T_s indicates the valid start time and T_e indicate valid time according to temporal data.

3.3 Simple Calendar Based Pattern

When temporal information is applied in terms of date, month , year and week they form the term calendar schema. It is introduced in temporal data mining. A calendar

schema is a relational schema (in the sense of relational databases) $R = (f_n : D_n, F_{n-1} : D_{n-1}, \dots, F_1 : d_1)$ together with a valid constraint. A calendar schema (year : {1995,1996,1997.....} , month : {1,2,3,4,.....12}, day : {1,2,3.....31} with the constraint is valid if that evaluates (yy, mm, dd) to True only if the combination gives a valid date. For example <1955,1,3> is a valid date while <,1996,2,31> is not.

In calendar pattern , the branch e cover e' in the same calendar schema if the time interval e' is the subset of e and they all follow the same pattern. If a calendar pattern $\langle d_n, d_{n-1}, d_{n-2}, \dots, d_1 \rangle$ covers another pattern $\langle d'_n, d'_{n-1}, d'_{n-2}, \dots, d_1 \rangle$ if and only if for each $I, 1 \leq i \leq n$ or $d_i = d'_i$. Now our task is to mine frequent pattern over arbitrary time interval in terms of calendar pattern schema.

4 Proposed Work

The support of dataset in the data warehouse can be maintained by dividing it into different intervals. The support of a item in interval t1 can not be the same in interval t2. A infrequent or less support item in interval t1 can be frequent item in interval t2.

The calendar schema is implemented by applying apriori algorithm [11]. It follows the candidate generation approach in order to mine the frequent item. We assist here that Total tree construction from partial tree is an efficient approach for mining time based associated items. It first constructs a partial tree (P tree). A P-tree is a set enumeration tree structure in which to store partial counts for item sets. The top, *single attribute*, level comprises an array of references to structures of the form shown to the right, one for each column [12]. Each branch indicate the association ship of item. It reduces the size of dataset and increases the performance and efficiency of algorithm. It can solve following queries (1) What are the frequent set over the interval t1 and t2 ? (2) what are the period when (a,b) item are frequent ? (3) Item which are dramatically change from t4 to t1.

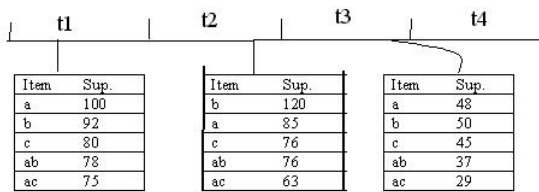


Fig. 1. Frequent pattern in different interval

4.1 Partial Support [2]

Most of the existing methods described above proceed essentially on each database pass by defining some candidate set and then examining each record to identify all the members of the candidate set that are subsets of the record, incrementing a support-count for each. The computational cost of this increases with the density of information in database records, i.e. when the average number of attributes present in

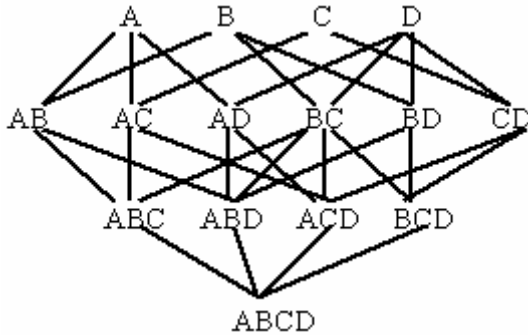


Fig. 2. Lattice of item {A,B,C,D}

a record is high, leading to an exponential increase in the number of subsets to be considered, and when candidate sets are large. In principle, however, it is possible to reduce this cost of subset-counting by exploiting the relationships between sets of items. For example, in the simplest case, a record containing the attribute set ABD will cause incrementation of the support-counts for each of the sets ABD , AB , AD , BD , A , B and D . Strictly, however, only the first of them is necessary, since a level of support for all the subsets of ABD can be inferred subsequently from the support-count of ABD .

Let i be a subset of the set I (where I is the set of n attributes represented by the database). We define P_i , the *partial support* for the set i , to be the number of records whose contents are identical with the set i . Then T_i , the *total support* for the set i , can be determined as:

$$T_i = \sum P_j (\forall j, j \supseteq i) \quad (1)$$

This allows us to postulate a general algorithm for computing total supports. Let P be the set of partial support counts P_i corresponding to sets i which appear as records in the database, and T be the set of total support counts in which we are interested (however this is defined). With the members of P and T initialized to zero.

Algorithm A

Inputs: Transaction DS , countset P

Output: Returns P and T counting sets in DS

Method:

```

A1:  $\forall$  Records  $j$  in  $DS$  do
    begin add 1 to  $P_j$ 
        insert  $j$  to  $P$ 
    end;

A2:  $\forall j$  in  $P$  do
    begin .  $i$  in  $T$ ,  $i \subseteq j$  do
        begin add  $P_j$  to  $T_i$ 
        end;
    end;
```

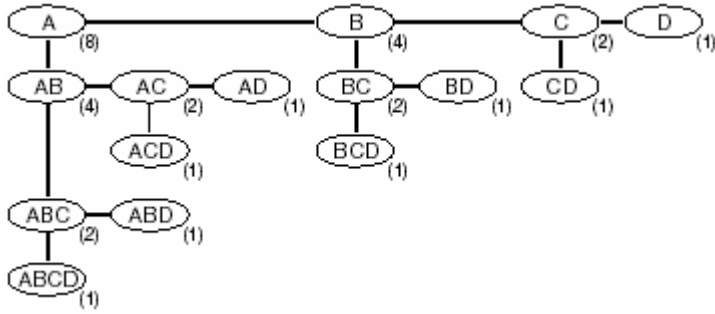


Fig. 3. Tree Storage of Subset (A,B,C,D)

For a database of m records, stage 1 of the algorithm (A1) performs m support-count incrementations in a single pass, to compute a total of m partial supports, for some $m' \leq m$. The second stage of the algorithm (A2) involves, for each of these, the examination of subsets which are members of the target set T . In an exhaustive version of the method, T will be the full set of subsets of I . Computing via summation of partial supports, however, offers three potential advantages. Firstly, when n is small ($2^n \ll m$), then A2 involves the summation of a set of counts, which is significantly smaller than a summation over the whole database. Secondly, even for large n , if the database contains a high degree of duplication ($m' \ll m$) then the stage 2 summation will again be significantly faster than a full database pass, especially if the duplicated records are densely-populated with attributes. Finally, and most generally, we may use the stage A1 to organize the partial counts in a way which will facilitate a more efficient stage 2 computation, exploiting the structural relationships inherent in the lattice of partial supports.

Figure 3 shows an alternative representation of the sets of subsets of I , for $I = \{A, B, C, D\}$, in the form of Rymon's [13] set enumeration tree. In this structure, each subtree contains all the supersets of the root node, which follow the root node in lexicographic order

4.2 Algorithm for Generating Calendar Based Temporal Association Rule Using TFP Tree

The proposed algorithm first extract the data of particular interval from whole data set and apply the TFP Mining approach to find frequent itemset on that specific intervals.

Input: A Transaction Database D, Specified calendar pattern <dd,mm,yy>

Output: Frequent item set, Temporal database table TDB

Method:

- Step (1) Set pointer to first record of database
- Step (2) Scan the Database one by one and follow the Step(3)
- Step (3) {

- Step (4) If $\text{data} \in \langle \text{dd}, \text{mm}, \text{yy} \rangle$
 Step (5) Send the data into Temporal database Table TDB
 Step (6) }
 Step (7) set $K = 1$
 Step (8) Build level K in the T -tree.
 Step (9) “Walk” the P -tree, applying algorithm TFP to add interim supports associated with individual P -tree nodes to the level K nodes established in (2).
 Step (10) Remove any level K T -tree nodes that do not have an adequate level of support.
 Step (11) Increase K by 1.
 Step (12) Repeat steps (8) through (11); until a level K is reached where no nodes are adequately Supported.

In above algorithm step (1) to step (5) used to find out the itemset, which occurs on valid time period specified by calendar schema. Step (7) to step (12) used for mining frequent itemset from TFP tree.

5 Experimental Observation

In this section we present the experimental result showing the performance of TFP tree approach and temporal TFP tree approach. The experiments were performed on the synthetic data set based on KDD cup T20I10D250kN500. The Pentium III with 128 MB Main Memory, 20 GB hard disk having Microsoft windows was used. Algorithms were implemented in C++ and Java. Figure 4(a) and 4(b) shows the comparative graph for different time intervals. The execution time taken by CPU in TFP algorithm is almost three times more than temporal TFP algorithm, which is a significant improvement and also shows that the performance of temporal TFP algorithm steadies as the support of itemset grows.

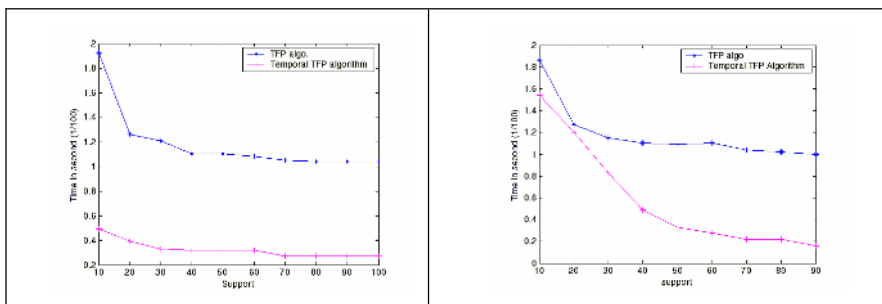


Fig. 4. Comparative graph of TFP tree and Temporal TFP tree

6 Conclusion and Future Work

In real world data the knowledge used for mining is always time varying. Real transactional databases specially exhibit temporal characteristic and time varying behavior. For example, in Telecommunication Data Analysis, the calling pattern may vary with time. Similarly for Market Basket Analysis the associations between various items may change with time and also that this transaction Database may have either a sparse or dense database. In this paper we have adopted time sensitive approach and presented an algorithm for mining frequent itemset on specified time interval using TFP approach. Frequent items generated for specific time interval have great impact from the databases. It has the capability to affect all aspects of doing business in today's world. It will empower decision makers with realistic results and that too with more accuracy and reduced time lag. It will thus help in more realistic and relevant decision-making. An inherent advantage of using P- tree and T- tree structure is in having branches which can be considered independently and therefore the structures can be rapidly adapted for use in parallel / distributed Association Rule Mining.

References

1. R. Agrawal and R. Srikant, R : Fast algorithm for mining association rule. In VLDB'94 Chile, Sept (1994).pp –487-499.
2. Coenen, F.P., Goulbourne, G. and Leng, P.H. (2001). Computing Association Rules Using Partial Totals. In de Raedt, L. and Siebes, A. (Eds), Principles of Data Mining and Knowledge Discovery, Proc PKDD 2001, Spring Verlag LNAI 2168, pp 54-66.
3. Frans Coenen, Paul Leng, and Shakil Ahmed, Data Structure for Association Rule Mining : T-tree and P- Tree, IEEE transaction on Knowledge Discovery and Data Engineering, Vol 16, No 6 ,(2004).
4. Chris Giannella_, Jiawei Hany, Jian Peiz, Xifeng Yany, Philip S. Yu R: Mining Frequent Patterns in Data Streams at Multiple Time Granularities, pg 191 – 210, H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha (eds.), Next Generation Data Mining, (2003).
5. Juan M .Ale , Gustavo H. Rossi R: An approach to discovering temporal association rules, ACM SIGDD March 1..21,(2002).
6. Banu Ozden , Sridhar Ramaswamy , Avi Silberschatz R : “Cyclic Association Rule” ,In Proc. Of fourteenth International conference on Data Engineering (1998), pp 412-425
7. Jian Pei, Jiawei Han, Yiwen Yin and Running Mao R : Mining Frequent Pattern without Candidate Generation Proc. ACM-SIGMOD Int'l Conf. Management of Data, pp. 1-12, (2000).
8. John F. Roddick, Kathleen Hornsby, Myra Spiliopoulou: An Updated Bibliography of Temporal, Spatial, and Spatio-temporal Data Mining Research. TSDM 2000: pp147-164.
9. Chris P. Rainsford, John F. Roddick R: Adding Temporal semantics to association rule, 3rd International conference KSS Springer 1999, pp 504-509
10. Xiandong Chen, Ilias Petrounian, Book “Knowledge Discovery and Data Mining” Chapter 5 “ A Development Framework of Temporal data Mining”, pp 93,2001.
11. Yingjiu Li, Peng Ning, X. Sean Wang , Sushil Jajodia. Discovering calendar- based temporal association rules, Data and Knowledge Engineering volume 4,Elsevier publisher, Volume 44 pp– 193-214 ,(2003).

12. Frans Coenen, Paul Leng, and Shakil Ahmed , Data Structure for Association Rule Mining : T-tree and P- Tree, IEEE transaction on Knowledge Discovery and Data Engineering, Vol 16, No 6 ,(2004).
13. R. Rymon,.,Search Through Systematic Set Enumeration, Proc. Third Int'l Conf. Principles of Knowledge and Reasoning, pp. 539-550, (1992).

Aquaculture Feature Extraction from Satellite Image Using Independent Component Analysis

JongGyu Han, KwangHoon Chi, and YeonKwang Yeon

Korea Institute of Geosciences & Mineral Resources,
305-350, 30 Gajung-dong Yusong-ku Daejeon, Republic of Korea
{jghan, khchi, ykyeon}@kigam.re.kr

Abstract. In multi-dimensional image, ICA-based feature extraction algorithm, which is proposed in this paper, is for the purpose of detecting target feature about pixel assumed as a linear mixed spectrum sphere, which is consisted of each different type of material object (target feature and background feature) in spectrum sphere of reflectance of each pixel. Landsat ETM+ satellite image is consisted of multi-dimensional data structure and, there is target feature, which is purposed to extract and various background image is mixed. In this paper, in order to eliminate background features (tidal flat, seawater and etc) around target feature (aquaculture) effectively, pixel spectrum sphere of target feature is projected onto the orthogonal spectrum sphere of background feature. The rest amount of spectrum sphere of target feature in the pixel can be presumed to remove spectrum sphere of background feature. In order to make sure the excellence of feature extraction method based on ICA, which is proposed in this paper, aquaculture feature extraction from Landsat ETM+ satellite image is applied. Also, In the side of feature extraction accuracy and the noise level, which is still remaining not to remove after feature extraction, we have conducted a comparing test with traditionally most popular method, maximum-likelihood. As a consequence, the proposed method from this paper can effectively eliminate background features around mixed spectrum sphere to extract target feature. So, we found that it had excellent detection efficiency.

1 Introduction

The image, which is obtained to take a picture for the surface of earth from Landsat ETM+ satellite is consisted of multi-dimensional data structure of multiplex spectrum sphere. A lot of data for the surface of earth are recorded such as an image of multi-dimensional data structure. In order to extract target feature from multiplex spectrum satellite image, there is an image processing method to change the peculiarity of data with the reflection of a certain axis for image data. At this time, the number of axis of reflected are needed to match the number of original data level so that the data level after reflection can be the same with the data before reflection or small to have the effect of compress.

There are methods to decide the axis to get the result as a meaningful outcome value when multi-dimensional data are reflected[1, 2]. ICA (Independent Component

Analysis) method, which is expanded method of PCA (Principal Component Analysis) is a statistical technique represented as a multi-dimensional vector of independence component with combination of linear. ICA can eliminate not only mutual relations of data but also higher level of mutual relations. In a consequence, it is the method to transform to independent between dimensions [3]. ICA is primarily applied to analyze data and extract feature. In this regards, we can use BSS (Blind Source Separation) method for original data and tracking down to find original data with mixed data without knowing the data to be mixed processing [4].

In this paper, ICA method is applied to extract wished feature data from multi-dimensional data structure. In this regard, Lee et al [6, 7], Hyvarinen [5] develops ICA-based feature extraction method based on the result of previous research works. An experiment about aquaculture feature extraction from Landsat ETM+ satellite image has been conducted to verify the validity of the result about actual application. In chapter 2, the materials and methods used in this paper are described and, in chapter 3, test result will be reviewed together with them. And, finally, the conclusion will be described in chapter 4.

2 Materials and Methods

This paper is described for algorithm of object detection, which can be classified as object and background from each pixels of multi spectral image consisted more than 2 object peculiarities. Multi spectrum image, which is used in this paper, is obtained from Landsat TM sensor with the following spectrum sphere: 450 nm~520 nm(band 1), 520 nm~ 600 nm(band 2), 630 nm~690 nm(band 3), 760 nm~ 900 nm(band 4), 1550 nm~1750 nm(band 5), 1040 nm~ 1250 nm(band 6), 2080 nm~2350 nm(band 7).

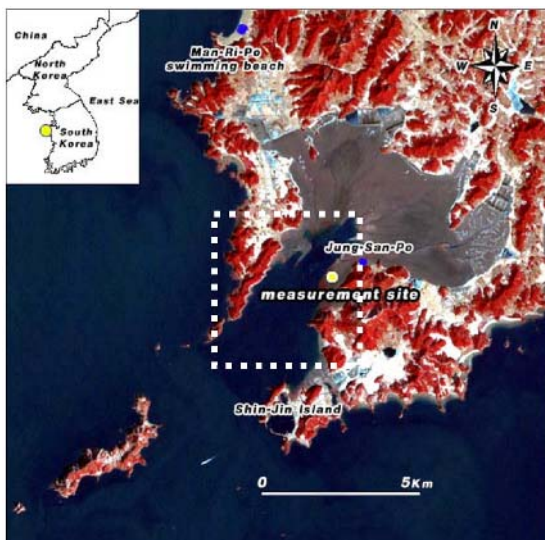


Fig. 1. Experiment data (Landsat ETM+) and research area

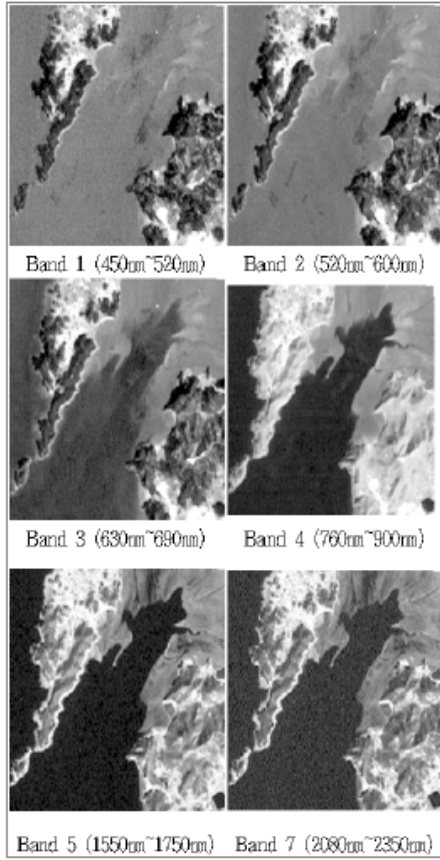


Fig. 2. Band image of different spectrum of LANDSAT ETM+ Satellite Image

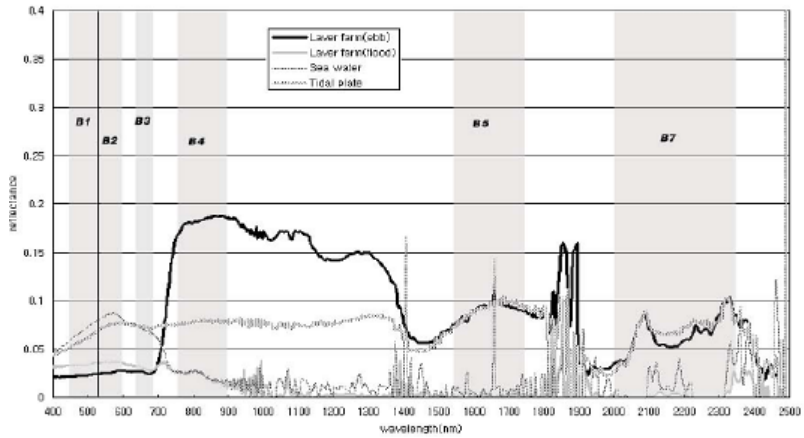


Fig. 3. Curved line of spectral reflection of Landsat ETM+ bands

In here, band 6, which is spectrum sphere of thermal infrared rays, is excluded in this paper. Fig. 1 shows you the satellite image of Landsat ETM+ used for experiment and, as a place of aquaculture; its location is Jungsanpo Taeahn-gun, Chungnam province. Fig. 2 shows you image data of each different spectrum band photographed from LANDSAT ETM+ Satellite Image around sea area of Jungsanpo Taean-gun, Chung-name province when ebb and flow is ebb tide on Feb 16, 2003.

Signal energy is controlled by reflected sunbeams radiant energy rather than discharged sunbeams by observed substance in spectrum sphere. Fig. 3 shows you the curved line of spectral reflectance, which is drawn with Landsat ETM+ spectrum sphere to measure spectral reflectance for aquaculture and around substances.

In algorithm for object detection of partial pixel based on mixed models of linear spectrum, reflected spectrum sphere of each pixel is presumed as linear mixed spectrum sphere from each different kinds of substance in the surface of pixel. If pixel spectrum sphere is controlled by object spectrum sphere, this pixel is the sign as an object pixel. If the contribution extent of object spectrum sphere can neglect in pixel spectrum sphere, that pixel can be classified as a background. As a consequence, what pixel spectrum sphere is disassembled as an organization spectrum sphere is needed to decide what kinds of substances are occupied extensively as a target or background.

3 Feature Extraction Procedure Using ICA

ICA is a technique to isolate mutual independent signals statistically from linear mixed signals and, it is applied to not only signal field but also image field vigorously. ICA method seeks for direction to reach independence elements in data.

There are lots of performance algorithms to exist for ICA such as Entropy minimized method, common data minimized method, Gaussian measure most suitable method and, maximum likelihood method. Lee[6, 7] is applied as a mixed model of independent element analyzer by maximum likelihood method. The weakness of this method is not extended easily to find a partial space. That weakness is not applied for fields like image coding. However, in an image segmentation and field of image classification, image is described as much smaller dimension than the numbers of pixels.

Fig. 4 is a sequence chart of data processing to extract target feature based on ICA method, which is indicated in this paper.

At first, in a data preparation stage as a first step, input feature with N unit from Landsat ETM+ Satellite Image data and creation of input data collection with $N+1$ dimension from output class. And, they go through the process of formalizing for each input features. In a second performing stage of ICA, ICA is performed to new data collections, which have been made in the data preparation step. And, the result is saved as an extra weight matrix W in dimension of $(N+1) \times (N+1)$.

In a third step, absolute average of each independent line vector, which has $N+1$ extra weight matrix W , is searched. And, among the extra weight line elements, elements value, which has an extra weight value less then absolute average, is made as zero. Fourth step is the stage of extracting for candidate feature. After all extra weight value line vectors are reflected in an original peculiar space, candidate feature of new $N+1$ unit is extracted to multiply original input data by new extra weight matrix of

$(N+1) \times N$ dimension. Extracted candidate features have been mixed with target and background features. Last stage is the step of deleting background feature from extracted candidate features. Candidate feature set F is made in this stage and, if extra weight value, which is copied with each candidate features, is zero, it is considered as background features and, it is removed. As a consequence, only final target features, which are removed background features from candidate features will be extracted.

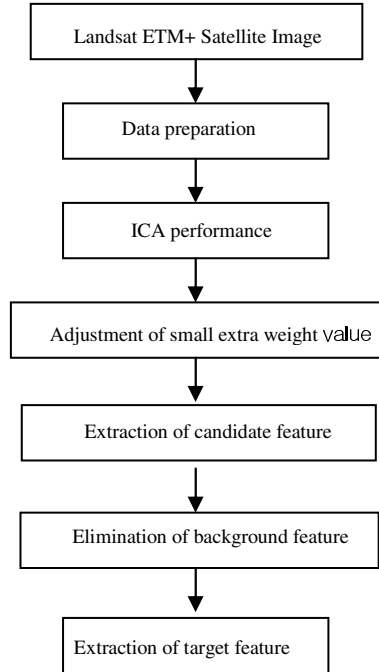


Fig. 4. The sequence chart for data processing of feature extraction

4 Test Results and Considerations

In this chapter, in order to extract aquaculture feature from Landsat ETM+ multi spectrum image data, image of 200×161 size is applied and, it is formalized to keep gaussian distribution peculiarity of $N(m, \sigma^2)$. Also, in order to delete noise effectively from original data, test has been conducted to find how features are extracted with keeping the average as regularly ($m=0$) and changing ($\sigma^2: 0 \sim 1$) the dispersion.

Fig. 5 shows you aquaculture feature extracted to apply feature extraction method based on ICA, which is proposed in this paper. Indicated section with black color in white dotted line is shown for aquaculture feature.

Fig. 6 shows you aquaculture feature extracted through distribution method of maximum likelihood, which is the most popular used. The test has been conducted by the side of noise, which is not eliminated after feature extraction and accuracy of extracted feature.

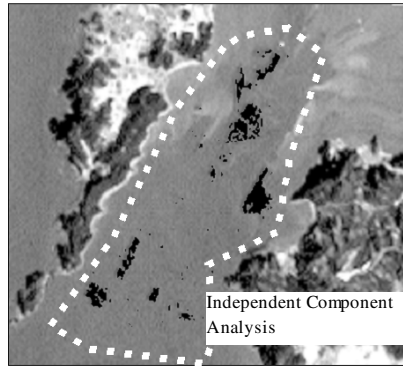


Fig. 5. Aquaculture feature through ICA-based feature extraction method (Black color in white dotted line)

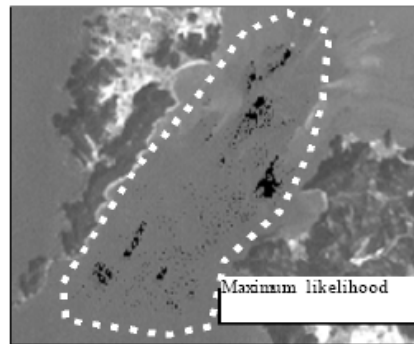


Fig. 6. Aquaculture feature extracted by maximum likelihood distribution method (Black color in white dotted line)

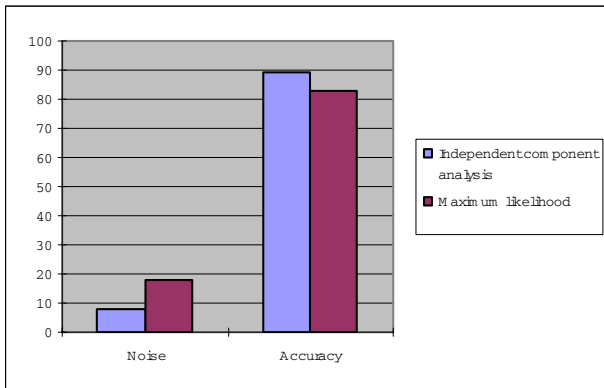


Fig. 7. Accuracy comparison for feature extraction and noise, which is not eliminated from extracted feature after test

Noise, which is not eliminated after feature extraction, from ICA-based feature extraction method is shown 10% lower than maximum likelihood method and, also, in the side of accuracy, ICA-based feature extraction method is shown 6% higher than maximum likelihood method. Test result of Fig. 5 and 6 are shown well in image and, it displays the superiority of ICA-based feature extraction method.

5 Conclusion and Discussion

ICA-based feature extraction algorithm, which is proposed in this paper, is purposed to detect target feature about pixel supposed as a linear mixed spectrum sphere, which is consisted of each different substance types (target feature and background feature) that has different reflection spectrum sphere of each pixels.

On Landsat ETM+ Satellite Image, which is consisted of multi dimensional data structure, ICA-based feature extraction method is indicated to eliminate background features (tidal flat, seawater and etc), which is located around target feature (aquaculture) effectively and, in order to confirm the superiority of proposed method, aquaculture feature can be successfully extracted on Landsat ETM+ satellite image.

In the side of noise level, which is not eliminated after feature extraction and accuracy, comparing test with maximum likelihood method, which is the most popular method traditionally, has been conducted.

As a consequence, the proposed method in this paper shows you superior detection performance in extraction of target feature to extract as background feature is eliminated effectively in mixed spectrum sphere around target feature.

References

1. P. J. Huber, Projection Pursuit, *The Annals of Statistics*, vol. 13, no. 2, pp. 435-475, 1985.
2. A. Hyvarinen, New Approximations of Differential Entropy for Independent Component Analysis and Projection Pursuit, *In Advances in Neural Information Processing Systems 10 (NIPS '97)*, pp. 273-279, 2003/02/18 22:55:40, 1998
3. A. Hyvarinen, J. Karhunen, and E. Oja, Independent Component Analysis, *John Wiley & Sons*, 2001.
4. A.J. Bell and T.J. Sejnowski, An Information-Maximization Approach to Blind Separation and Deconvolution, *Neural Computation*, vol. 7, pp. 1129-1159, 1995.
5. A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626-634, 1999.
6. T.-W. Lee, M. Girolami, and T.J. Sejnowski. Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources. *Neural Computation*, 11(2):417-441, 1999.
7. T.-W. Lee, M.S. Lewicki, and T.J. Sejnowski. Unsupervised classification with non-Gaussian mixture models using ICA. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems 11, Cambridge, MA, 1999. NIPS, MIT Press*.

Modeling the Organoleptic Properties of Matured Wine Distillates

S.B. Kotsiantis¹, G.E. Tsekouras², C. Raptis³, and P.E. Pintelas¹

¹ Educational Software Development Laboratory, Department of Mathematics,
University of Patras, Greece

{sotos, pintelas}@math.upatras.gr

² Department of Cultural Technology and Communication,
University of the Aegean, Mytilene, Greece

gtsek@ct.aegean.gr

³ S&E&A METAXA Distilleries S. A., Athens, Greece
craptis@metaxa.com.gr

Abstract. We present how the supervised machine learning techniques can be used to predict quality characteristics in an important chemical engineering application: the wine distillate maturation process. A number of experiments have been conducted with six regression-based algorithms, where the M5' algorithm was proved to be the most appropriate for predicting the organoleptic properties of the matured wine distillates. The rules that are exported by the algorithm are as accurate as human expert's decisions.

1 Introduction

There are many chemical engineering processes, where the quality characteristics of the product cannot be measured objectively either on-line due to the lack of proper sensors or off-line due to the absence of any measuring devices [11]. In these cases, a human expert is employed to assign the product quality characteristics to certain pre-defined categories (classes), based on his experience and perceptions. The procedure of employing a human expert to perform the classification usually requires the interruption of the process in order to collect a sample. Furthermore, this way of classifying the product quality is very subjective and may lead to significant errors, especially when the same expert is not always employed to perform the classification [2], [7].

A different approach to classify quality parameters is to use supervised algorithms in order to automate the process. In this paper, we present how the supervised machine learning techniques can be used to predict quality characteristics in an important chemical engineering application: the wine distillate maturation process.

The case under study is a part of the maturation process of METAXA Distilleries, a Greek aged wine distillates producing company. The firm has the problem of blending aged distillates to produce a series of final products of different quality specifications. Since most distillates characteristics cannot be accurately modelled as a function of their organoleptic properties, a system that anticipates automatically this relationship is of at most importance for the blending process engineer.

The following section describes in brief the problem and the dataset of our study. In section 3 we present the basic design issues of the supervised machine learning techniques that are used here. Section 4 compares the experimental results obtained by these techniques, while the concluding remarks are given in section 5.

2 Problem and Data Description

Freshly distilled spirits such as wine distillates, whiskies and rums have pungent, unpleasant odorous and sharp taste. The organoleptic properties are improved by storing the distillates in oak barrels for several years. During this process, which is known as maturation process, a number of wood components are extracted and many chemical reactions take place. However, due to the plethora of factors that affect the maturation process such as immature distillate, size, nature and usage of the barrel, environmental conditions etc., the maturation mechanisms are not completely understood. Furthermore, there is no reliable chemical or physical index that can indicate the progress of the maturation process.

The case under study is a part of the maturation process of S&E&A METAXA Distilleries S. A., a famous Greek aged wine distillates producing company. Until today, an expert tries some samples from the barrels and carries out the product quality classification, based on his perceptions. It is clear that a system, which can automatically anticipate the organoleptic properties of the distillates, based on some other accurately measured distillate characteristics, is of main importance. We applied supervised machine learning algorithms to develop models for the classification of the aroma and taste of the distillate. The intensity of aroma is of relative importance since it is basically a measure of quantity. Persistence of aroma is an indication of quality, particularly in the lingering bouquet of a mature wine.

For the aroma and taste prediction, we are based on the following input parameters:

- x_1 =barrel usage (the number of refills of each barrel)
- x_2 =barrel age (in years)
- x_3 =distillate age (in years)

The available data consisted of 170 input–output pairs [11]. The output values in the data set are the classifications for the aroma (y_1) and taste (y_2), which were given by the expert using discrete values ranging from 0 to 10, with a step of 1, where 0 and 10 correspond to the worst and finest quality, respectively. This dataset reflects long years of knowledge and experience about the process and consequently, until now it is usually used for the suggestion of suitable distillates to obtain a consistent blend from one production batch to the next.

Given ordered classes, one is not only interested in maximizing the classification accuracy, but also in minimizing the distances between the actual and the predicted classes. The usage of regression algorithms to solve ordinal classification problems has been examined in [5]. In this case each class needs to be mapped to a numeric value. Another approach is to reduce the multi-class ordinal classification problem to a set of binary classification problems using the one-against-all approach [4]. Because the problem can be solved either with regression techniques or ordinal classification techniques we present both techniques in the next section.

3 Supervised Machine Learning Techniques

The problem of regression consists in obtaining a functional model that relates the value of a target continuous variable y with the values of variables x_1, x_2, \dots, x_n (the predictors). This model is obtained using samples of the unknown regression function. These samples describe different mappings between the predictor and the target variables.

For the propose of our comparison the six most common regression techniques namely Model Trees and Rules [12], Neural Networks [6], Linear regression [3], Locally weighted linear regression [1] and Support Vector Machines [10] are used. In the following we will briefly describe these regression techniques.

Linear regression (LR) is the simplest statistical technique used to find the best-fitting linear relationship between the class and its predictors (other features).

$$y = \beta_0 + \beta_1 x_{i1} + \dots + \beta_k x_{ik}$$

Find values of beta that minimize Q :

$$Q = \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik}))^2$$

Note that nominal features with n values are converted into $n-1$ binary features and a Wald test is used to test the statistical significance of each coefficient (β_j) in the model [3].

Model trees are the counterpart of decision trees for regression tasks. Model trees are trees that classify instances by sorting them based on attribute values. Instances are classified starting at the root node and sorting them based on their attribute values. The most well known model tree inducer is the M5' [12]. A model tree is generated in two stages. The first builds an ordinary decision tree, using as splitting criterion the maximization of the intra-subset variation of the target value [13]. The second prunes this tree back by replacing subtrees with linear regression functions wherever this seems appropriate.

M5rules algorithm produces propositional regression rules in IF-THEN rule format using routines for generating a decision list from M5' Model trees [13]. The algorithm is able to deal with both continuous and nominal variables, and obtains a piecewise linear model of the data.

Artificial Neural Networks (ANNs) are another method of inductive learning based on computational models of biological neurons and networks of neurons as found in the central nervous system of humans [6]. Regression with a neural network takes place in two distinct phases. First, the network is trained on a set of paired data to determine the input-output mapping. The weights of the connections between neurons are then fixed and the network is used to predict the numerical class values of a new set of data. Back Propagation (BP) is the most well known technique for training ANNs.

Locally weighted linear regression (LWR) is a combination of instance-based learning and linear regression [1]. Instead of performing a linear regression on the full, unweighted dataset, it performs a weighted linear regression, weighting the training instances according to their distance to the test instance at hand. This means that a linear regression has to be performed for each new test instance, which makes the

method computationally quite expensive. However, it also makes it highly flexible, and enables it to approximate non-linear target functions.

The sequential minimal optimization algorithm (SMO) has been shown to be an effective method for training support vector machines (SVMs) on classification tasks defined on sparse data sets [9]. SMO differs from most SVM algorithms in that it does not require a quadratic programming solver. In [10] SMO is generalized so that it can handle regression problems (SMOreg). This implementation globally replaces all missing values and transforms nominal attributes into binary ones.

As we have previously mentioned the presented problem can be also solved by ordinal classification techniques. The most sophisticated approach that enables standard classification algorithms to make use of ordering information in ordinal class attributes is presented in [4]. This method converts the original ordinal class problem into a series of binary class problems that encode the ordering of the original classes. However, to predict the class value of an unseen instance this algorithm needs to estimate the probabilities of the k original ordinal classes using our $k - 1$ models. For example, for a three class ordinal problem, estimation of the probability for the first ordinal class value depends on a single classifier: $\Pr(\text{Target} < \text{first value})$ as well as for the last ordinal class: $\Pr(\text{Target} > \text{second value})$. Whereas, for class value in the middle of the range, the probability depends on a pair of classifiers and is given by

$$\Pr(\text{Target} > \text{first value}) * (1 - \Pr(\text{Target} > \text{second value})).$$

4 Experiments Results

All accuracy estimates were obtained by averaging the results from 10 separate runs of stratified 10-fold cross-validation. In cross-validation technique, the training set is divided into mutually exclusive and equal-sized subsets and for each subset the regressor is trained on the union of all the other subsets. An estimation of the regressor’s criterion is then the average of the error rate of each subset.

It must be mentioned that we mainly used the free available source code for our experiments by the book [13]. For our problem the regression criteria are most suitable. However, there isn’t only one regressor’s criterion. Table 1 represents the most well known. Fortunately, it turns out that in most practical situations the best regression method is still the best no matter which error measure is used.

Table 1. Regressors’ criteria (p : predicted values, a : actual values, $\bar{a} = \frac{1}{n} \sum_i a_i$)

Mean absolute error	$(p_1 - a_1 + \dots + p_n - a_n) / n$
Root mean squared error	$\sqrt{(p_1 - a_1)^2 + \dots + (p_n - a_n)^2} / n$
Relative absolute error	$(p_1 - a_1 + \dots + p_n - a_n) / (a_1 - \bar{a} + \dots + a_n - \bar{a})$
Root relative squared error	$\sqrt{(p_1 - a_1)^2 + \dots + (p_n - a_n)^2} / \sqrt{(a_1 - \bar{a})^2 + \dots + (a_n - \bar{a})^2}$

In Table 2, the regressors' criteria for each algorithm for aroma estimation are presented.

Table 2. Aroma

	<i>M5'</i>	<i>BP</i>	<i>LR</i>	<i>LWR</i>	<i>SMO-reg</i>	<i>M5rules</i>	<i>Ordinal technique</i>
Mean absolute error	0.45	0.83	0.78	0.72	0.76	0.46	0.50
Root mean squared error	0.56	1.01	0.93	0.86	0.97	0.56	0.67
Relative absolute error	22.48	40.93	38.52	35.71	37.21	22.69	24.70
Root relative squared error	22.16	40.27	36.86	34.11	37.95	22.34	26.71

In Table 3, the regressors' criteria for each algorithm for taste estimation are presented.

Table 3. Taste

	<i>M5'</i>	<i>BP</i>	<i>LR</i>	<i>LWR</i>	<i>SMO-reg</i>	<i>M5rules</i>	<i>Ordinal technique</i>
Mean absolute error	0.62	0.88	0.76	0.75	0.75	0.64	0.80
Root mean squared error	0.79	1.10	0.96	0.95	0.98	0.81	0.98
Relative absolute error	25.31	36.09	31.23	30.85	31.02	26.29	32.64
Root relative squared error	28.06	39.37	34.42	33.75	35.02	29.08	34.72

distillateAge <= 3.33 : LM1 (51/23.386%)

distillateAge > 3.33 :

| distillateAge <= 8 : LM2 (86/25.419%)

| distillateAge > 8 : LM3 (33/20.786%)

LM num: 1

FinalScore = -0.1313 * barrelUsage - 0.0594 * barrelAge + 1.6481 * distillateAge + 2.0491

LM num: 2

FinalScore = -0.2139 * barrelUsage - 0.0479 * barrelAge + 0.4957*distillateAge + 6.2112

LM num: 3

FinalScore = -0.1134 * barrelUsage - 0.0705 * barrelAge + 0.5631 * distillateAge + 6.1581

Fig. 1. M5' model tree for aroma prediction

According to the results, the M5' is the most accurate algorithm to be used for our problem. An advantage of M5' except for its better performance is its comprehensibility. In figure 1, we present the produced rules for the prediction of wine aroma by the M5' algorithm.

In figure 2, we present the produced rules by the M5' algorithm for the prediction of wine taste.

```

distillateAge <= 6.67 :
| distillateAge <= 1.68 : LM1 (27/21.7%)
| distillateAge > 1.68 :
| | barrelUsage <= 2.5 : LM2 (22/38.291%)
| | barrelUsage > 2.5 : LM3 (64/28.848%)
distillateAge > 6.67 : LM4 (57/21.572%)

LM num: 1
FinalScore = -0.1829 * barrelUsage - 0.1107 * barrelAge + 1.3845 * distillateAge + 2.7629

LM num: 2
FinalScore = -0.1837 * barrelUsage - 0.1958 * barrelAge + 0.9566 * distillateAge + 5.1988

LM num: 3
FinalScore = -0.2414 * barrelUsage - 0.1428 * barrelAge + 0.6525*distillateAge + 4.8879

LM num: 4
FinalScore = -0.1417 * barrelUsage - 0.1032 * barrelAge + 1.0884*distillateAge + 1.7945

```

Fig. 2. M5' model tree for taste prediction

It must be mentioned that the exported rules by the algorithm are as accurate as human experts' decisions.

5 Conclusion

Food processing is most often characterized by severe complexity, non-linearity and lack of objective information regarding the qualitative final product characteristics. The increasing and strong need for total quality management in food industries has rendered the construction of flexible and robust automotive decision making systems for product evaluation.

It is long been recognized that the classification of aged wine distillates is a non-linear, multi-criteria decision making problem characterized by great complexity, non-linearity and lack of objective information regarding the desired final product qualitative characteristics. The most efficient solution for the evaluation of aged wine distillates estimations with emphasis on aroma and taste, when an appropriate mathematical model cannot be incorporated, is to develop adequate and reliable expert systems based on machine learning for the classification.

In this paper, we presented how the supervised machine learning techniques can be used to predict the quality characteristics of matured wine distillates. Six algorithms were applied and compared each other with respect to real-life data, taken from a wine distillates producing company. The results showed that the M5' algorithm was the most appropriate, among the tested algorithms, for predicting the organoleptic properties of the distillates. The rules that are exported by the algorithm are as accurate as human experts' decisions.

Using machine learning in wine industry has clear advantages: it does not color the score of an individual wine with a tester's bias and it retains a level of objectivity that allows comparisons across all available wines. In a future work we will use supervised machine learning techniques to classify the variety of the wine as well as the production place (origin denomination). This classification can be carried out by processing information corresponding to physical features (color, density, conductivity, etc.) and chemical features (phenols, anthocians, amino acids, etc) [8].

References

1. Atkeson, C. G., Moore, A.W., & Schaal, S., Locally weighted learning. *Artificial Intelligence Review*, 11, (1997) 11–73.
2. Bende, M., & Nordin, S., Perceptual learning in olfaction: Professional wine tasters versus controls. *Physiology & Behavior*, 62, (1997) 1065–1070.
3. Fox, J., *Applied Regression Analysis, Linear Models, and Related Methods*, ISBN: 080394540X, Sage Pubns (1997).
4. Frank, E. and Hall M.: A simple approach to ordinal prediction, L. De Raedt and P. Flach (Eds.): *ECML 2001, LNAI 2167*, pp. 145-156, (2001), Springer-Verlag Berlin.
5. Herbrich R., Graepel T., and Obermayer K.: *Regression models for ordinal data: A machine learning approach*. Technical report, TU Berlin, (1999).
6. Mitchell, T., *Machine Learning*. McGraw Hill (1997).
7. Parr, W. V., Heatherbell, D., & White, K. G., Demystifying wine expertise: Olfactory threshold, perceptual skill and semantic memory in expert and novice wine judges. *Chemical Senses*, 27, (2002) 747–755.
8. Pena-Neira, A. I., Hernandez, T., Garcia-Vallejo, C., Estrella, I., & Suarez, J., A survey of phenolic compounds in Spanish wines of different geographical origins. *European Food Research and Technology*, 210, (2000) 445–448.
9. Platt, J., Using sparseness and analytic QP to speed training of support vector machines. In: Kearns, M. S., Solla, S. A. & Cohn D. A. (Eds.), *Advances in neural information processing systems 11*. MA: MIT Press (1999).
10. Shevade, S., Keerthi, S., Bhattacharyya C., and Murthy, K., Improvements to the SMO algorithm for SVM regression. *IEEE Transaction on Neural Networks*, 11(5), (2000) 1188–1183.
11. Tsekouras, G., Sarimveis, H., Raptis, C., Bafas, G., A fuzzy logic approach for the classification of product qualitative characteristics, *Computers and Chemical Engineering* 26 (2002) 429–438.
12. Wang, Y. & Witten, I. H., Induction of model trees for predicting continuous classes, In *Proc. of the Poster Papers of the European Conference on ML, Prague* (pp. 128–137). Prague: University of Economics, Faculty of Informatics and Statistics (1997).
13. Witten, I.H., Frank, E., *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, San Mateo, CA, (2000).

Bagging Random Trees for Estimation of Tissue Softness

S.B. Kotsiantis¹, G.E. Tsekouras², and P.E. Pintelas¹

¹ Educational Software Development Laboratory,
Department of Mathematics, University of Patras, Greece
{sotos, pintelas}@math.upatras.gr

² Department of Cultural Technology and Communication,
University of the Aegean, Mytilene, Greece
gtsek@ct.aegean.gr

Abstract. We present an ensemble of classifiers that can be used to predict quality characteristics of an important process in pulp and paper industry: the tissue softness estimation. This classification problem is a difficult one since, with respect to our data set, the accuracy of all the well-known classifiers is below 68%. Contrary to that, the bagging random trees ensemble model is able to increase the accuracy up to 75%.

1 Introduction

A process in pulp and paper industry is the tissue softness estimation. The production of tissue paper, softness is one the key attributes to improving the marketability of the final product. Consistent production of soft tissue is the key in ensuring production; sales and cost targets are met.

A problem related to tissue softness estimation is that the quality characteristics of the final product cannot be objectively measured due to the lack of proper sensors and measuring devices [10]. Thus, in order to estimate the quality, an expert is usually employed to carry out the classification by assigning the tissue softness characteristics to certain predefined categories. However, in most of the cases the use of an expert to perform the above classification requires the interruption of the process in order to collect samples. Another issue is that the classification of the quality is subjective and may lead to errors, especially when the same expert is not always employed to perform the classification.

In this paper we use supervised machine learning algorithms to automatically determine the tissue softness characteristics. The implementation of various well-known classifiers to the available data set yielded accuracy below 68%. This fact directly implies that the process is a very difficult one. To increase the accuracy and to improve the classification efficiency we use a technique called bagging random decision trees, which can provide more accurate results. After a number of experiments we showed that, in contrast to the most well known classifiers and ensemble methods, the used technique can significantly increase the classification accuracy of the process.

The following section describes in brief the problem and the dataset of our study. In section 3 we present the basic design issues of the supervised machine learning techniques. In section 4, we present the used ensemble method. Section 5 compares the experimental results obtained by other well-known techniques with the used ensemble, while the concluding remarks are given in section 6.

2 Problem and Data Description

Consumer acceptance of tissue is strongly influenced by the level of softness. Softness is a complex human perception that is influenced by both physical properties and psychological factors. Both bulk softness and surface softness are factors in the overall perception of softness.

Tissue is used mainly for body care, so quality issues in tissue making are more important than in any other type of paper production. There are several parameters, which affect the tissue softness, including the constitution and consistency of pulp, pulp refining and thickness of stock in the paper machine. However, the most important parameters are the ones related to the drying process, which is designed so that the sheet detaches from the Yankee dryer prior to contact with the doctor blade and forms a loop (the microfold) [6]. These parameters are the doctor blade angle and wear, the crepe ratio (the ratio of the speed of the sheet on the Yankee dryer to the speed of the sheet on the reel) and the rates of the chemicals, which adjust coating and release. Due to this process, the produced tissue becomes softer, bulkier and more absorbent. Tissue softness is usually measured in a subjective manner by the touch of an expert who compares it with some scaled samples.

For this reason, a model that infers tissue softness from other variables, which are accurately measured, could be of great use. A model using supervised machine learning techniques can be developed for a tissue-producing process, using the tissue softness as the output variable and the following parameters as input variables:

- x1: cross directional tensile of tissue
- x2: machine directional tensile of tissue
- x3: machine directional stretch of tissue
- x4: Yankee coating rate (ml/min)
- x5: Yankee release rate (ml/min)

The available data consisted of 375 input–output pairs. More information about data set can be found in [18].

3 Machine Learning Techniques and Estimation of Tissue Softness

Supervised machine learning is the exploration for algorithms that reason from externally supplied instances to produce general hypotheses, which will make predictions about future instances. In other words, the goal of supervised learning is to build a concise model of the distribution of the class label in terms of the predictor features. The resulting classifier is then used to assign class labels to the testing instances

where the values of the predictor features are known but the value of the class label is unknown.

Decision trees are trees that classify instances by sorting them based on attribute values. Each node in a decision tree represents an attribute in an instance to be classified, and each branch represents a value that the node can take. A recent overview of existing work in decision trees is provided in [14]. In rule induction systems, a decision rule is defined as a sequence of Boolean clauses linked by logical AND operators that together imply membership in a particular class [10]. The general goal is to construct the smallest rule-set that is consistent with the training data.

Artificial Neural Networks (ANNs) are another method of inductive learning and they all based on computational models of biological neurons [13]. A multi layer neural network consists of large number of units (neurons) joined together in a pattern of connections. First, the network is trained on a set of paired data to determine the input-output mapping. The weights of the connections between neurons are then fixed and the network is used to determine the classifications of a new set of data.

Naive Bayes classifier is the simplest form of Bayesian network [7]. This algorithm captures the assumption that every attribute is independent from the rest of the attributes, given the state of the class attribute. The assumption of independence is clearly almost always wrong. However, a large-scale comparison of Naive Bayes classifier with state-of-the-art algorithms on standard benchmark datasets found it sometimes to be superior to each of the other learning schemes [7].

Instance-based learning algorithms belong in the category of lazy-learning algorithms [13], as they delay the induction process until classification is performed. One of the most straightforward instance-based learning algorithms is the nearest neighbour algorithm [1]. K-Nearest Neighbour (kNN) assumes that the instances within a data set will generally exist in close proximity with other instances of the similar class.

The SVM technique revolves around the notion of a ‘margin’ that separates two data classes. Maximizing the margin, and thereby creating the largest possible distance between the separating hyperplanes can reduce the upper bound on the expected generalization error [5]. However, most real-world problems involve non-separable data for which no hyperplane exists that successfully separates the positive from negative instances in the training set. The solution is then to map the data into a higher-dimensional space and define a separating hyperplane there.

For the purpose of this study, a representative algorithm for each described learning technique was used. The most commonly used C4.5 algorithm [17] was the representative of the decision trees in our study. The most well known learning algorithm to estimate the values of the weights of a neural network - the Back Propagation (BP) algorithm [13] - was the representative of the ANNs. The Naive Bayes algorithm, which we used, is based on estimating marginal Gaussian estimators for numerical attributes. The 3-NN algorithm that combines robustness to noise and less time for classification than using a larger k for kNN was also used [1]. Finally, the Sequential Minimal Optimization (or SMO) algorithm was the representative of the SVMs as one of the fastest methods to train SVMs [16].

All accuracy estimates were obtained by averaging the results from stratified 10-fold cross-validation in our dataset. It must be mentioned that we used the free avail-

able source code for our experiments by the book [12]. As one can see, this classification problem is very difficult since the accuracy of all the well known classifiers in this domain is below 68% according to our dataset.

Table 1. Accuracy of simple models in our dataset

	NB	C4.5	3NN	BP	RIPPER	SMO
Accuracy	54.93%	67.20%	60.53%	58.93%	64.27%	60%

For this reason, there is a need for a more accurate technique. Combining classifiers is proposed as a new direction for the improvement of the classification accuracy [2]. When multiple classifiers are combined using voting methodology, we expect to obtain good results based on the belief that the majority of experts are more likely to be correct in their decision when they agree in their opinion. Voters can express the degree of their preference using a confidence score i.e. the probabilities of classifiers prediction.

4 Used Technique

Random Forests, recently introduced by Breiman [4], have been shown to be a powerful classification technique. In bagging, single models are induced over bootstrap samples of the training data, and the classification is made by voting. Random Forests is a particular implementation of bagging in which each model is a Random Tree. In a Random Tree for each split, rather than considering all possible splits, a tournament is held with a small group of randomly selected splits, and the best split of this smaller group is chosen. Therefore randomness enters the Random Forest algorithm in two places: the bootstrap sample for each tree, and choice of splits for participation in tournament selection. Random Forests exhibit many desirable characteristics: parallelism, high accuracy, fast to train, built-in error predictor, and a tendency not to overfit [4].

In the case of Random Forests, tree decorrelation is attained through randomness in choosing the bootstrap sample to train each tree, and tournament selection in growing each tree. As tournament size F approaches the total number of possible splits in the training set, the procedure becomes equivalent to bagging decision trees (where each tree is grown deterministically). Any other value of F corresponds to Random Forests.

In our implementation, instead of select possible splits at random, evaluate them and choose best, we evaluate all splits, and choose at random one of the best. This modification would be expected to slightly increase the correlation between individual trees. We also do not use pruning, as the bagging-like procedure can only reduce the variance, but not improve on high bias [2]. Therefore one probably want a low bias learner, and no pruning means lower bias.

Finally, the used technique is presented in Fig. 1.

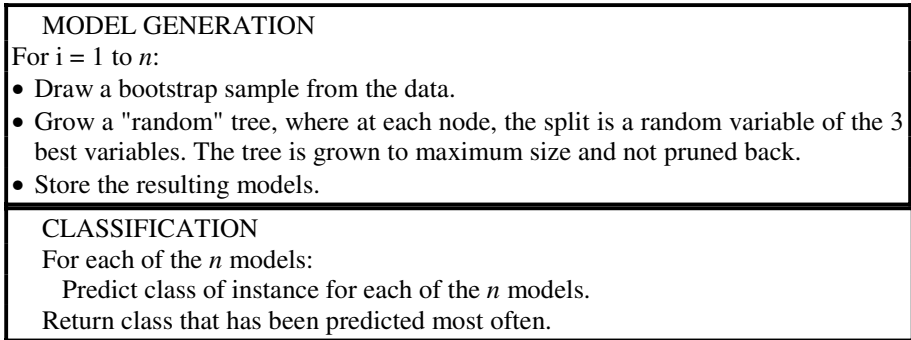


Fig. 1. The used ensemble

The used ensemble has a free parameter: the numbers of models (n). In our experiments, we used 90 models since most of the researchers used 100 models for their ensembles [2], [16], [20] and we wanted this technique to use less time for training. It must be also mentioned that the used ensemble can be easily parallelized.

The computations required to obtain the classifiers in each bootstrap sample are independent of each other. Therefore we can assign tasks to each processor in a balanced manner. By the end each processor has obtained a part of the ensemble. In the case we use the master-slave parallel programming technique, the method starts with the master splitting the work to be done in small tasks and assigning them to each slave. Then the master performs an iteration in which if a slave returns a result (this means it finished its work) then the master assigns it another task if there are still tasks to be executed. Once all the tasks have been carried out the master process obtains the results and orders the slaves to finish since there are not more tasks to be carried out. This parallel execution of the presented ensemble can achieve almost linear speedup.

In the following section, we briefly describe the most well known ensembles techniques and we compare the used technique with the other well known ensembles.

5 Experiments Results in Relation to Other Ensembles Techniques

Boosting [8] is similar in overall structure to bagging, except that keeps track of the performance of the data mining algorithm and concentrates on instances that have not been correctly learned. Instead of choosing the t training instances randomly using a uniform distribution, it chooses the training instances in such a manner as to favor the instances that have not been accurately learned. AdaBoost is a practical version of the boosting approach [8]. It was subsequently observed that Adaboost is in effect approximating a stagewise additive logistic regression model by optimising an exponential criterion [9]. This leads us to new variants of Adaboost that fit additive models directly. One such variant is Logitboost, which uses the Newton-like steps to optimise the loss criterion [16].

MultiBoosting [20] is another method that can be considered as wagging committees formed by AdaBoost. Wagging is a variant of bagging; bagging uses resampling to get the datasets for training and producing a weak hypothesis, whereas wagging uses reweighting for each training example, pursuing the effect of bagging in a different way. Another meta-learner (DECORATE, Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples) is presented in [12] that uses a learner to build a diverse committee. This is accomplished by adding different randomly constructed examples to the training set when building new committee members. Finally, LMT method builds logistic model trees, which are classification trees with logistic regression functions at the leaves [11].

Firstly, we compared the used technique with the previous referred well known ensembles. All accuracy estimates were obtained by averaging the results from stratified 10-fold cross-validation in our data set. The t-test was also used to statistically compare the algorithms. Throughout, we speak of two results for the dataset as being "significant different" if the difference is statistical significant at the 1% level according to the corrected resampled t-test [15], with each pair of data points consisting of the estimates obtained in one of the 100 folds for our method and the other ensemble being compared.

The presented ensemble is significantly more accurate than Bagging C4.5 with 100 models, Boosting C4.5 with 100 models, Boosting NB with 100 models, Random Rorest with 100 trees and Decorate C4.5 with 100 iterations, according to the corrected resampled t-test [15] (see Table 2). We used 100 models for the ensembles methods such as [2].

Table 2. Comparing the used technique

	Bagging random trees	Bagging C4.5 (100 models)	Boosting C4.5 (100 models)	Boosting NB (100 models)	Random Rorest (100 trees)	Decorate C4.5
Accuracy	74.40%	69.60%	69.60%	62.67%	69.07%	69.07%

Moreover, the presented ensemble is significantly more accurate than Multiboost C4.5 with 100 models, Logitboost DS with 100 models, Bagging BP with 100 models, Boosting BP with 100 models and LMT classifier, according to the corrected resampled t-test [15] (see Table 3).

Table 3. Comparing the used technique

	Bagging random trees	Multiboost C4.5 (100 models)	Logitboost DS (100 models)	Bagging BP (100 models)	Boosting BP (100 models)	LMT
Accuracy	74.40%	69.60%	60.27%	61.07%	58.67%	68.27%

To sum up, the performance of the presented ensemble is more accurate than the other well-known classifiers and ensembles. The used ensemble can achieve an increase in classification accuracy from 11% to 35% compared to single learners. Moreover, the average relative accuracy improvement of the used methodology is from 7% to 26% in relation to the well known ensembles techniques. This indicates that it is possible to obtain a feasible solution to the problem with the used technique.

6 Conclusion

Tissue is a true consumer product and as such there are continuous demands to improve quality and performance. One of the most commonly sought after improvements is the softness of the tissue. Tissue softness is perceived by the consumer as the primary quality property.

In our case study, the classification of tissue softness was a very difficult problem since the accuracy of all the well known classifiers in this domain was below 68% according to our dataset. Contrary to that, the bagging random trees ensemble model was able to increase the accuracy up to 75%. In a following work, we will try to further increase the classification accuracy.

References

1. Aha, D., *Lazy Learning*. Dordrecht: Kluwer Academic Publishers (1997).
2. Bauer, E. & Kohavi, R., An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36, (1999) 105–139.
3. Breiman L., Bagging Predictors. *Machine Learning*, 24(3), (1996) 123-140.
4. Breiman, L., Random Forests. *Machine Learning* 45 (1), (2001) 5-32.
5. Burges, C., A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2, (1998) 1-47.
6. Corbooy, W. G., Yankee dryers. In B. A. Thorp, Paper machine operations. In: *Pulp and paper manufacture*, vol. 7. GA:Joint Textbook Committee of the Paper Industry (1991).
7. Domingos, P. and Pazzani, M., On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29, (1997) 103-130.
8. Freund Y. and Schapire R. E., Experiments with a New Boosting Algorithm, *Proceedings: ICML'96*, p. 148-156.
9. J. H. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28 (2000) 337 – 374.
10. Furnkranz, J., Separate-and-Conquer Rule Learning, *Artificial Intelligence Review*, Vol. 13, (1999) 3-54.
11. N.Landwehr, M.Hall, E. Frank, Logistic Model Trees, 14th European Conference on Machine Learning ECML (2003) 241-252.
12. P. Melville and R. Mooney, Constructing Diverse Classifier Ensembles using Artificial Training Examples, *Proc. of the IJCAI-2003*, pp.505-510, Mexico, August 2003
13. Mitchell, T., *Machine Learning*. McGraw Hill (1997).
14. Murthy, S., Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. *Data Mining and Knowledge Discovery*, 2 (1998) 345–389.

15. Nadeau, C., Bengio, Y., Inference for the Generalization Error. *Machine Learning* 52(3), (2003) 239-281.
16. Platt, J., Using sparseness and analytic QP to speed training of support vector machines. In: Kearns, M. S., Solla, S. A. & Cohn D. A. (Eds.), *Advances in neural information processing systems* 11. MA: MIT Press (1999).
17. Quinlan, J. R., *C4.5: Programs for machine learning*. Morgan Kaufmann, San Francisco (1993).
18. Tsekouras, G., Sarimveis, H., Raptis, C., Bafas, G., A fuzzy logic approach for the classification of product qualitative characteristics, *Computers and Chemical Engineering* 26 (2002) 429–438.
19. Witten, I.H., Frank, E., *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, San Mateo, CA, (2000).
20. Webb, G. I., *MultiBoosting: A Technique for Combining Boosting and Wagging*, *Machine Learning*, 40, (2000) 159–196, Kluwer Academic Publishers.

Concept Mining for Indexing Medical Literature

Isabelle Bichindaritz and Sarada Akkineni

University of Washington, 1900 Commerce Street, Box 358426,
Tacoma, WA 98402, USA
ibichind@u.washington.edu

Abstract. This article addresses the task of mining concepts from biomedical literature to index and search through this documents base. This research takes place within the Telemakus project, which has for goal to support and facilitate the knowledge discovery process by providing retrieval, visual, and interaction tools to mine and map research findings from research literature in the field of aging. A concept mining component automating research findings extraction such as the one presented here, would permit Telemakus to be efficiently applied to other domains. The main principle that has been followed in this project has been to mine from the legends of the documents the research findings as relationships between concepts from the medical literature. The concept mining proceeds through stages of syntactic analysis, semantic analysis, relationships building, and ranking.

1 Introduction

As the number of scientific publications is increasing tremendously, it is becoming hard for researchers to find relevant information in a domain and keep up with the information flow. Researchers are foremost interested in collecting and analyzing the research findings presented in research literature. There is a need for an information system that can be used by researchers to extract domain knowledge in the form of research findings without trying to read the whole article. The idea pursued in this article is to extract knowledge from articles using mostly tables and figures legends, since this is where the main research findings are the most likely to be represented.

With the rapidly growing body of scientific knowledge and increasing overspecialization in specific domains, it is likely that the scientific work of one research group might also solve an important problem that arises in the work of another group. Yet the two groups might not be aware of the work of each other. However, important knowledge is recorded at least in textual form in bibliographic databases, such as Medline for the field of biomedicine. In the present context these large documents databases provide both an opportunity and a need for developing advanced methods and tools for computer supported knowledge discovery [2, 3].

The goal of literature-based discovery in general is to discover new and potentially meaningful relations between a given starting concept of interest and other concepts by mining bibliographic databases [5, 6, 7] such as Medline [13]. The idea of discovering new relations from a bibliographic database was introduced by [13, 14], who

describes a data mining system that made seven medical discoveries that have been later published in relevant medical journals.

Telemakus is one such information system dedicated to the idea of literature-based information mining [1] for a core set of concepts in a medical domain. The motivating idea of Telemakus project is to aid researchers in the caloric restriction and aging domain to rapidly find the research findings and main concepts from research articles [8] in this domain. The research findings are represented by relationships between pairs of such concepts. The system can then be used to find, in particular, all the articles that study a particular relationship. For instance, a researcher can use this system to find all the articles which studied the relationship between caloric restriction and aging. The most helpful knowledge any researcher can gain while entering a new domain is the set of research findings so far, which is what can be more easily acquired using Telemakus system [4]. Telemakus project is intended to assist researchers by providing visual and interaction tools to visualize and navigate the research findings in their domain of research [4].

The system presented here proposes to automate the process of extracting the research findings from the literature by mining the concepts and relationships from documents and indexing these by the concepts learnt. An important feature of this system is that it mines for relationships between concepts, such as the relationship between caloric restriction and aging, and not for isolated concepts. The next section presents the Telemakus project. The third section sets forth the system architecture, and the fourth section explains the concept mining process in detail through its different stages. It is followed by the results and a conclusion.

2 Telemakus Project

The goal of the Telemakus project at the University of Washington Medical School is to support and facilitate the knowledge discovery process by developing interaction tools to retrieve and visualize documents by their research findings. For that purpose, this system mines and maps research findings from research literature. Telemakus system proposes tools and a framework to create, maintain, and query a database of documents through their research findings. Telemakus is part of SAGE KE, the Science of Aging Knowledge Environment and an online resource for researchers in the field of aging [15].

The Telemakus system [4] consists of a set of domain documents (current focus is the biology of aging), a conceptual schema to represent the main components of each document, and a set of tools to query, visualize, maintain, and map the set of documents through their concepts and research findings [4]. The conceptual schema is composed of standard bibliographic information, information about the research process (age, sex, number of subjects, treatment regimen, and research criteria for research animals), and most importantly research findings derived from data tables and figures. The “Unified Medical Language System” (UMLS), a specialized knowledge source in biomedicine, provides standardized concepts for the creation of a controlled domain vocabulary. The UMLS provides a very powerful resource for rapidly creating a robust scientific thesaurus in support of precision searching. Further, the semantic type descriptors for each concept and semantic network may offer some interesting oppor-

tunities for intelligent searching and mapping of concepts representing research findings, and their relationships. At present, knowledge extraction resorts to systems with both manual and automated components. A key area of current work is to move towards automating the research concept identification process, through data mining [4].

Fig. 1 shows the architecture of Telemakus system. *Fetcher*, *Extractor*, and *Cross-check* are the most important components in the system. *Fetcher* fetches documents from the bibliographic database and stores them in the domain-specific database (referred to as DSDB). The extraction process is performed by a domain expert who manually interprets each legend to extract relevant concepts and uses his or her knowledge of the domain to establish relationships.

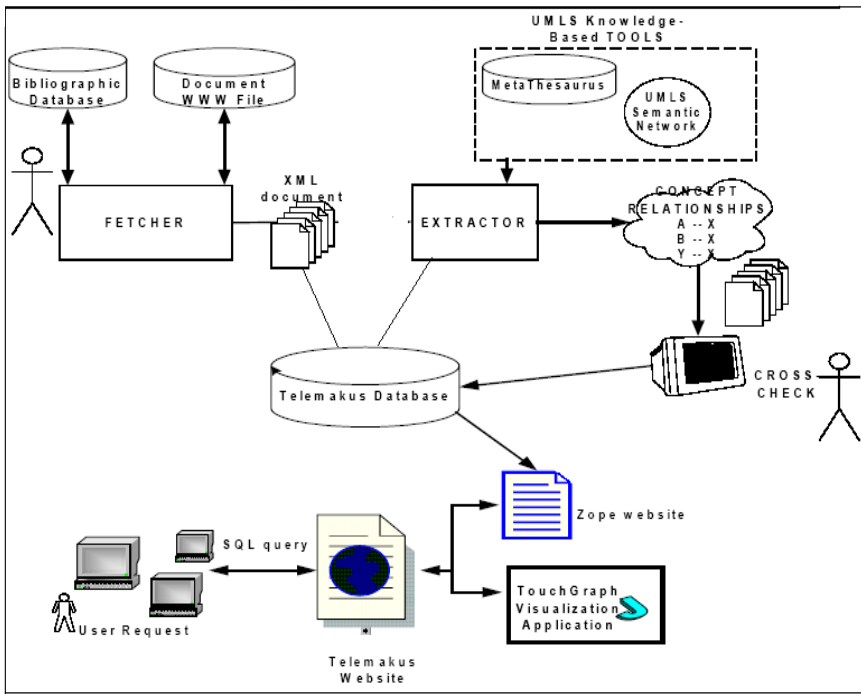


Fig. 1. Telemakus architecture

Crosscheck is a visualization tool used by researchers through the Web interface and by the domain expert for evaluation. The current Telemakus database has a Web interface, which can be accessed at www.Telemakus.net.

3 Architecture

The architecture of the systems (see Fig. 2) shows the datastores and components of the concept mining system. There are two knowledge bases involved, UMLS data-

base, and DSDB database. Within DSDB, the domain specific thesaurus represents the standardized vocabulary of the “caloric restriction and aging” domain. The components of the system are the following:

1. *Data Access Component*, which extracts the document to mine from DSDB.
2. *Syntactic Analyzer*, which analyzes the syntax of a document by parsing its legends and extracting lexical information.
3. *Relationship Builder*, which takes the lexical information from above, locates a trigger phrase for a relationship from each legend, and forms from there a triple composed of two phrases and a trigger phrase. Each triple represents a relationship between two concepts.
4. *Relationship Selector*, which semantically analyzes each phrase in each triple by accessing the UMLS, and extracts from each phrase its main concepts.
5. *Ranker*, which ranks the relationships extracted.
6. *Evaluator*, which evaluates the result of the concept mining process for each document. It accesses DSDB through Data Access Component in order to get the results of the manual mining process, and compare them with the automated ones. It produces precision and recall ratios as the reference in evaluating the system.

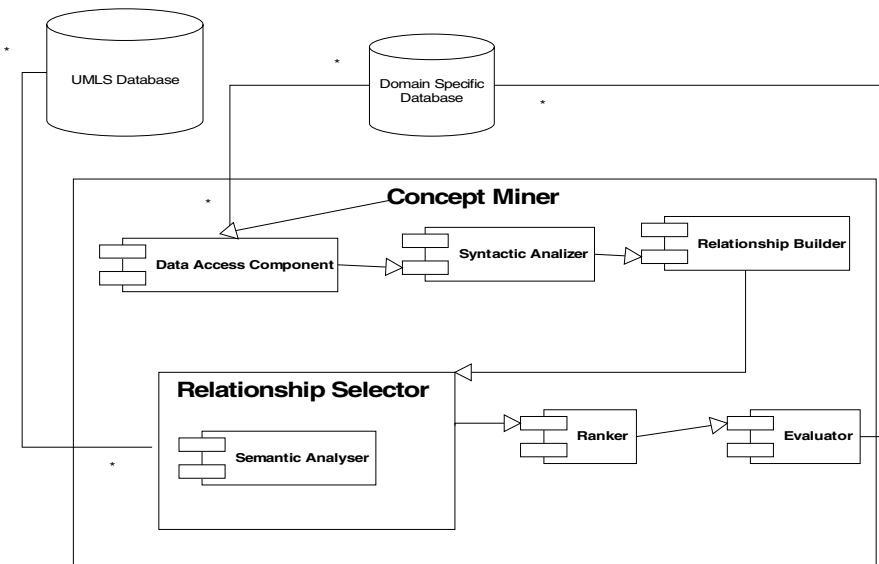


Fig. 2. Concept Miner architecture

4 Concept Mining Process

Concept mining involves processing articles already stored in domain specific database (DSDB). These articles currently do not comprise the full text of the original

articles, only the sections of documents that are of most interest. These are tables and figure descriptions, referred to as *legends*, which are considered the most probable placeholders for research findings. The full text of the article is not provided as input to the system. It has been established by Telemakus project team that the most interesting information about research literature is usually found in legends [15].

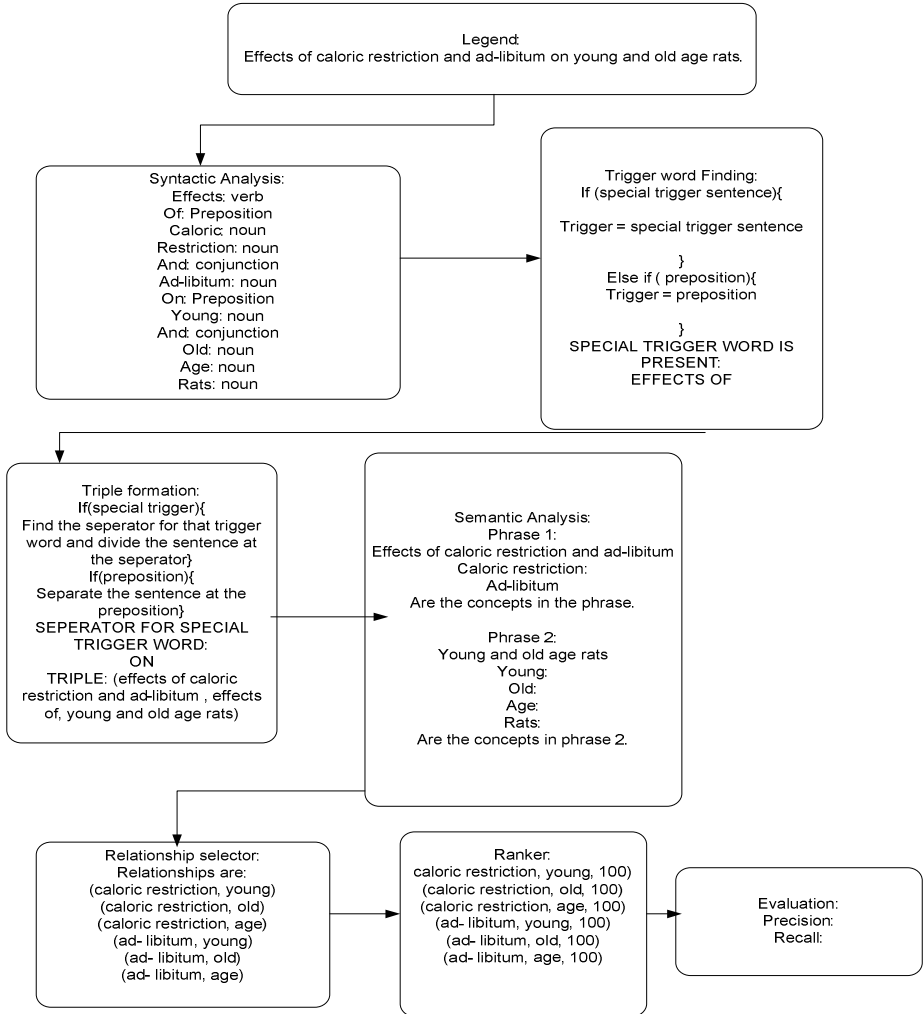


Fig. 3. Concept Miner system process flow

Given an article or a set of articles, the system starts by extracting all legends already stored in the database, processes each legend by identifying interesting relationships, filters relationships, ranks those relationships based on a number of parameters, and

finally writes the resulting relationships to an XML file for later use. For comparison purposes, precision and recall are also computed by the system on a per-article basis.

The concept mining process can be divided into three main phases, *syntactic analysis*, *semantic analysis* and *concept mapping and association*.

4.1 Syntactic Analysis

In a given document, there may be one or more legends. Each legend may contain one or more sentences. For any given legend, the text is first broken into constituent sentences. The process of concept extraction and association is applied at the sentence level. Each sentence is parsed and grammatical structures are extracted. From the concept association perspective, each sentence is made up of a connector phrase, called a *trigger phrase*, and the two phrases connected by that trigger phrase. An example of trigger phrase shown on Fig. 3 is “effects of”. These trigger phrases are usually prepositions, but human experts have also provided special phrases that act as triggers, such as “effect of”. A trigger phrase may contain a connector phrase that separates the remaining part of the sentence into two phrases. After a trigger is found in a sentence, the remaining sentence is split into two phrases optionally connected by a connector phrase. This phase of the system is called *syntactic analysis* in a broad sense. The connector word and two phrases together are called a *triple*.

Syntactic parsing determines the structure of the sentence being analyzed. Syntactic analysis involves parsing the sentence to extract information contained within word ordering. Syntactic parsing is computationally less expensive than semantic processing. Syntactic analyzers can be classified into two types depending on their approach to analysis. *Top-down analyzer* starts from the root symbol of the grammar and successively predicts (usually in a left-to-right manner) what its constituent parts should be. On the other hand, *bottom-up analyzer* starts from the string to be analyzed and attempts to construct a syntactic tree by recognizing the right-hand sides of the grammar rules and thus reducing those portions of the string to their non-terminals. This process is repeated until the only remaining non-terminal is the root symbol of the grammar.

For the purpose of this project, the *bottom-up analyzer* is the best fit based on the input pattern, which means that the input string should be taken as input and parsed. Currently, the system uses a basic parser API called *Specialist Text Tools* API that is an open source java implementation [11]. This parser is a minimal commitment barrier category parser. The minimal commitment analysis assigns underspecified syntactic analysis to lexically analyzed input. The parser package contains a shallow parser that extracts minimal phrases from sentences. Using the *Specialist lexicon*, the part of speech and other syntactic information are analyzed. This analysis is specific to biomedical field. The *Specialist Text Tools* tokenizer package tokenizes text into words, sentences, and sections. It can handle free text and Medline citation formats. Sentences are found by looking for sentence bounding punctuation for the most part, and looking at the capitalization of the next word that follows. This method is not always successful, particularly when abbreviations such as *Dr.* and *Mr.* are met. A list of acronyms is consulted when periods are hit. Two new lines in a row are also considered a sentence break. By the end of processing, an analyzed sentence contains all the tokens that make up the sentence, along with their character offsets back to the original document. The results of this phase are a set of *triples* (see Fig. 3 for an example).

4.2 Semantic Analysis

After triples are built, each triple is further analyzed by *semantic analysis*. This involves looking for concepts in each phrase, and is accomplished by applying a domain specific natural language processing tool. From each phrase, a candidate list of concept phrases from the UMLS is extracted. The semantic analysis is made possible by the National Library of Medicine (NLM)'s UMLS [12]. UMLS ultimate goal is to facilitate the development of computer systems that behave as if they "understand" the meaning of the language of biomedicine and health.

Although the *Specialist Text Tools* also resort to UMLS, it is a foremost knowledge source for semantic analysis. The words or phrases are considered as concepts in the medical domain if said words or phrases can be found in the metathesaurus of the UMLS. Metathesaurus is one of the three UMLS knowledge sources. It is the central vocabulary component of the UMLS [12].

In this project, initially, UMLS metathesaurus entries have been extracted manually in order to create a consistent controlled base vocabulary specific to the domain of caloric restriction and nutritional aspects of aging, and placed within DBSB database. These UMLS selected thesauri characterize mainly research concepts and process description, such as organism type. As new concepts are identified from the documents' tables and figures, they are translated into their UMLS preferred terms, which are added to the controlled vocabulary database.

Semantic analysis is performed on the results of syntactic analysis of the legends to determine the meaning of the words in the sentence. In this step, the semantics of each word or phrase is evaluated. Though there are a few choices for performing semantic analysis on free text, this project uses *MMTx* tool [10] as it is specifically developed for the biomedical field. The main purpose of *MMTx* semantic analysis is to find out the phrases and their variants and then match these to the phrases or words in the UMLS database. The words or phrases successfully mapped to the UMLS database can be considered as concepts in the biomedical or health field. The concept mapping process can be summarized as follows (see Fig. 3 for an example):

- Parse the text into noun phrases and perform the remaining steps for each phrase;
- Generate the variants for the noun phrase where a variant essentially consists of one or more noun phrase words together with all of its spelling variants, abbreviations, acronyms, synonyms, inflectional and derivational variants, and meaningful combinations of these;
- Form the candidate set of all Meta strings containing one of the variants;
- For each candidate, compute the mapping from the noun phrase and calculate the strength of the mapping using an evaluation function. Order the candidates by mapping strength;
- Combine candidates involved with disjoint parts of the noun phrase, recompute the match strength based on the combined candidates;
- Select those having the highest score to form a set of best Meta mappings for the original noun phrase.

4.3 Concept Mapping and Association

Semantic analysis produces a candidate list of concepts for each phrase. This list is refined in multiple steps. First, duplicates or substrings that may be referring to the same concept are removed at the sentence level. For example, two candidate concepts “muscle mass increase” and “muscles” are considered as same candidate concepts. The latter is removed from the list since the algorithm favors the most precise concepts. Next, a known synonym transformation is performed. This step is necessary to replace generic candidate concept names with those preferred by the domain. For example, “free access to food” is replaced with “ad libitum”.

To improve precision, this list is further filtered by pattern matching the concept variations. A preliminary list of relationships is created from these two candidate concept lists of left and right phrases. This is the *relationship selector* phase. General approach to association is based on trigger phrase in a sentence. Each sentence is conceptually equivalent to a list of left-side candidates and a list of right-side candidates connected by a trigger phrase. From these lists, a set of relationships is constructed. Each relationship is a candidate association, containing exactly one left-side concept and one right-side concept (see Fig. 3 for an example).

At the article level, all unique relationships from constituent sentences are aggregated. This list is again refined to remove partial matches. This step is necessary to remove partial matches that may be unique at the sentence level but partially equivalent at the article level.

This list of relationships is ranked based on the importance of concepts, in particular based on the presence of the concepts in the domain-specific database. They are further ranked based on the weight of the relationships involved. The weight of a relationship is the sum of the weights of concepts in that relationship. The weight of a concept is calculated as the number of times this concept occurs in a relationship.

After obtaining a short list of relationships, the effectiveness of the system is evaluated by precision and recall. Finally, the results are written to an XML file, and made available to other parts of the Telemakus system, like Crosscheck.

5 Results

This system is designed to analyze multiple documents at the same time. The success of the system is determined by the recall and precision ratios. Current results show an average recall of 81% and precision of 50% for partial match. Precision is the ratio of matching relations to the total number of relations identified. Recall is the ratio of matching relations to the total number of relations identified by the manual process. The precision and recall are calculated in two ways: partial matching and total matching. In partial matching strategy, if the system extracted relationship (muscle mass – caloric restriction) and the manual results has relationship (muscle mass increase – caloric restriction), then this relationship is considered a match. In total matching, the relationship should be present in the manual results exactly matching both concepts.

The system is evaluated for 30 random articles. The average values of recall and precision for these 30 documents are shown in Table 1. It shows that the average values of precision and recall are much higher when partial matches of the concepts

Table 1. Precision and recall ratios

Number of Documents	Total Recall	Total Precision	Partial Recall	Partial Precision
Total	53%	35%	81%	50%

are also considered as a match. The reason for considering partial matching is that, there can be some implied knowledge that is used by the domain expert during the manual process, but that kind of knowledge is either not available to this system or hard to automate. These results are encouraging because relationships mining is a much more complex task, in particular when it involves semantic analysis such as in this system, than classical information retrieval. Some of the relationships retrieved do not even share a word with sentences in the documents. These figures compare system performance with human performance, while even humans between themselves would not retrieve the same relationships. An interesting result is that the relationships extracted make good sense, even though the human indexer may not have selected these as the most important in a document.

6 Conclusion

Mining concepts to index literature is a complex task that requires many levels of refinements at both the syntactic and semantic level. Although the results so far are encouraging, several ways are still open for refinements. In the future, the algorithm to extract relationships can be improved in particular by eliminating the legends that represent facts, instead of research findings. Another improvement will be to store the legends in the database in full, so that the input to the automated system and the manual system are the same. This system approach is very original in comparison with other systems mining biomedical literature for concepts, because it mines for relationships between concepts, and not for isolated concepts. Telemakus indexes and navigates through the documents database by relationships between concepts.

Acknowledgements

We want to thank Sherrilynne Fuller, Debra Revere, and Paul Bugni, from the Division of Biomedical and Health Informatics of the University of Washington for providing the idea, the data, and their support role throughout this work.

References

1. Chang, C., Hsu C.: Enabling concept-based relevance feedback for information retrieval on the WWW. Knowledge and Data Engineering (IEEE) vol.11 issue 4 (1999) 595-609
2. Dorre, J., Gerstl, P., Seiffert, R.: Text mining: finding nuggets in mountains of textual data. In: Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM press (1999) 398-401

3. Friedman, C., Kra, P., Yu, H., Krauthammer, M., Rzhetsky, A.: GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles, *Bioinformatics* 17 Suppl 1 (2001) S74-S82
4. Fuller, S., Revere, D., Bugni, P., Martin, G.M.: A knowledgebase system to enhance scientific discovery: Telemakus. *Biomed Digit Libr.* Sep 21;1(1):2 (2004)
5. Hand, D., Mannila, H., Smyth, P.: Principles of data mining. MIT Press (2001)
6. Hearst, M.A.: Untangling Text Data Mining. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, University of Maryland (1999) 3-10
7. Jiawei, H., Micheline, K.: Data mining concepts and techniques. 1st edn. Morgan Kaufmann (2000)
8. Lin, S., Chen, M.C., Ho, J., Huang, Y.: ACIRD: Intelligent Internet Document Organization and Retrieval. *IEEE Transactions on Knowledge and Data Engineering* vol. 14 (2002) 599-614
9. Nasukawa T., Nagano, T.: Text Analysis and Knowledge Mining System. Knowledge management Special Issue. *IBM systems journal* Vol. 40 (2001) 967-984
10. National Library of Medicine: MetaMap Transfer (MMTx). (2005) <http://mmtx.nlm.nih.gov> [Last access: 2005-04-01]
11. National Library of Medicine: The Specialist NLP Tools. (2004) <http://specialist.nlm.nih.gov> [Last access: 2005-04-01]
12. National Library of Medicine: The Unified Medical Language System. (2005) <http://umls.nlm.nih.gov> [Last access: 2005-04-01]
13. Swanson, D.R.: Information discovery from complementary literatures: Categorizing viruses as potential weapons. *Journal of the American Society for Information Science* Vol. 52(10) (2001) 797-812
14. Swanson, D.R., Smalheiser, N.R.: An interactive system for finding complementary literatures: a stimulus to scientific discovery. *Artificial Intelligence* Vol.9 (1997) 183-203
15. Telemakus project team: Mining and Mapping Research Findings to Promote Knowledge Discovery (2001) <http://www.telemakus.net/papers.html> [Last access: 2005-04-01]

Author Index

- Adam, Sébastien 194
Adegorite, Adeoye I. 466
Ahmad, Khurshid 570
Akkineni, Sarada 682
Alhaji, Reda 346
Almeida Neto, Areolino de 295
- Baesens, Bart 80
Bagherjeiran, Abraham 120
Bagirov, Adil 62, 71
Bahi, Halima 507
Bak, EunSang 275
Barbu, Eugen 194
Barker, Ken 346
Basir, Otman A. 466
Bhatnagar, Vasudha 11
Bichindaritz, Isabelle 682
Bobrowski, Leon 218
Bouguila, Nizar 42
Boullé, Marc 228, 253, 600
Bousquet, Olivier 100
Bunke, Horst 366
- Candillier, Laurent 100
Cao, Wenming 305
Chen, Huiping 549
Chen, Xiaoping 487
Chi, KwangHoon 660
- Deinzer, Frank 415
Derichs, Christian 415
Dickinson, Peter 366
Dong, Jian-xiong 590
Du, Jun 346
- Eick, Christoph F. 120
- Ferrandiz, Sylvain 253, 600
Fevens, Thomas 314
- Ghosh, Moumita 62, 71
Ghosh, Ranadhir 62, 71
Giacinto, Giorgio 184
Gillam, Lee 570
Graveron-Demilly, D. 325
Grigat, Rolf-Rainer 437
Gupta, Anamika 11
- Halvey, Martin 174
Hamano, Shinichi 641
Hammouda, Khaled M. 265
Han, JongGyu 660
Haraguchi, Makoto 537
Haralick, Robert 132
Harpaz, Rave 132
Hayashi, Akira 356
Héroux, Pierre 194
Homaeian, Leila 334
Hu, Jianhui 305
Huang, Tianqiang 549
Huynh, Van-Nam 516
Huysmans, Johan 80
- Imiya, Atsushi 497
Irniger, Christophe 366
- Jänichen, Silke 153
- Kamel, Mohamed S.
265, 466, 580
Karras, D.A. 325
Keane, Mark T. 174
Kim, E. 1
Kinoshenko, Dmitry 445
Klusch, Matthias 610
Ko, J. 1
Korkmaz, Emin Erkan 346
Kotsiantis, S.B. 667, 674
Kovalev, Vassili 456
Kraetzl, Miro 366

- Krasotkina, Olga 52
 Krödel, Michael 405
 Krzyżak, Adam 314, 590
 Kudo, Mineichi 90
 Kuhlmann, Annette 558
 Kuhnert, Klaus-Dieter 405
 Kumar, Naveen 11
 Kurgan, Lukasz 334
- Lai, J.Y. 426
 Latecki, Longin Jan 476
 Laur, Pierre-Alain 395
 Lazarescu, Mihai M. 163
 Le, Cuong Anh 516
 Legrand, Gaëlle 203
 Li, Shao 630
 Li, Shuo 314
 Li, Song 314
 Lu, Naijiang 487
 Lübbling, Christoph 558
 Luebke, Karsten 110
- Makrehchi, Masoud 580
 Mashtalir, Vladimir 445
 Matute, Diego N. 265
 Mertzios, B.G. 325
 Miezianko, Roland 476
 Miura, Takao 376
 Mizuhara, Yuko 356
 Mottl, Vadim 52
 Muchnik, Ilya 52
- Nakamura, Atsuyoshi 90
 Nicoloyannis, Nicolas 203
 Niemann, Heinrich 415
 Nikulin, Vladimir 142
 Nock, Richard 395
- Ohnishi, Naoya 497
 Okubo, Yoshiaki 537
 Oliveira, Alexandre Cesar Muniz de
 285
- Paiva, Anselmo Cardoso de
 285, 295
 Perdisci, Roberto 184
- Perner, Petra 153
 Petrou, Maria 456
 Piater, Justus 243
 Pintelas, P.E. 667, 674
 Pokrajac, Dragoljub 476
 Poncelet, Pascal 395
 Ponson, Dominique 590
 Pray, Keith A. 384
- Raptis, C. 667
 Roli, Fabio 184
 Rouhana, Alain 120
 Ruiz, Carolina 384
- Sato, Masako 641
 Scalzo, Fabien 243
 Sellami, Mokhtar 507
 Seredin, Oleg 52
 Shaban, Khaled B. 466
 Sharma, L.K. 620
 Shimazu, Akira 516
 Shimizu, Kazuhiro 376
 Sia, William 163
 Silva, Aristófanés Corrêa
 285, 295
 Silva, Josenildo C. da 610
 Silva, Valdeci Ribeiro da, Junior
 295
 Smola, Alex J. 142
 Smyth, Barry 174
 Sowmya, A. 426
 Suematsu, Nobuo 356
 Suen, Ching Y. 590
 Sun, Yanmin 21
 Sy, Bon K. 526
 Symphor, Jean-Emile 395
 Szepannek, Gero 110
- Takigawa, Ichigaku 90
 Taniguchi, Tsuyoshi 537
 Tao, Li 549
 Tellier, Isabella 100
 Thole, Clemens-August 558
 Tiwary, U.S. 620
 Torre, Fabien 100

Trinder, J. 426
Trupin, Eric 194
Tsekouras, G.E. 667, 674

van Ormondt, D. 325
Vanthienen, Jan 80
Verma, Keshri 651
Vetter, Ralf-Michael 558
Vilalta, Ricardo 120
Vinarsky, Vladimir 445
Vyas, O.P. 620, 651
Vyas, Ranjana 620, 651

Wang, Jiandong 549
Wang, Shoujue 305
Wang, Yang 21
Weihs, Claus 110

Weng, Shifeng 630
Wong, Andrew K.C. 21
Wu, Shiliang 549

Xia, Yinglong 630
Xiang, Shiming 487
Xiao, Gang 305

Ye, Feiyue 549
Yearwood, John 62, 71
Yegorova, Elena 445
Yeon, YeonKwang 660

Zhang, Baibo 31
Zhang, Changshui 31, 487, 630
Zhao, Shuyan 437
Ziou, Djemel 42