

Petra Perner (Ed.)

LNAI 7377

Advances in Data Mining

Applications and Theoretical Aspects

12th Industrial Conference, ICDM 2012
Berlin, Germany, July 2012
Proceedings

 Springer

Lecture Notes in Artificial Intelligence 7377

Subseries of Lecture Notes in Computer Science

LNAI Series Editors

Randy Goebel

University of Alberta, Edmonton, Canada

Yuzuru Tanaka

Hokkaido University, Sapporo, Japan

Wolfgang Wahlster

DFKI and Saarland University, Saarbrücken, Germany

LNAI Founding Series Editor

Joerg Siekmann

DFKI and Saarland University, Saarbrücken, Germany

Petra Perner (Ed.)

Advances in Data Mining

Applications and
Theoretical Aspects

12th Industrial Conference, ICDM 2012
Berlin, Germany, July 13-20, 2012
Proceedings

 Springer

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editor

Petra Perner
Institute of Computer Vision
and Applied Computer Sciences, IBAI
Kohlenstraße 2
04107 Leipzig, Germany
E-mail: pperner@ibai-institut.de

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-31487-2 e-ISBN 978-3-642-31488-9
DOI 10.1007/978-3-642-31488-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012940664

CR Subject Classification (1998): I.2.6, I.2, H.2.8, J.3, H.3, I.4-5, J.1

LNCS Sublibrary: SL 7 – Artificial Intelligence

© Springer-Verlag Berlin Heidelberg 2012

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.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The 12th event of the Industrial Conference on Data Mining ICDM was held in Berlin (www.data-mining-forum.de) running under the umbrella of the World Congress “The Frontiers in Intelligent Data and Signal Analysis, DSA 2012” (www.worldcongressdsa.com).

For this edition the Program Committee received 97 submissions. After the peer-review process, we accepted 32 high-quality papers for oral presentation of which 22 are included in this proceedings book. The topics range from theoretical aspects of data mining to applications of data mining such as in multimedia data, marketing, finance and telecommunications, medicine and agriculture, and process control, industry and society. Extended versions of selected papers will appear in the *International Journal Transactions on Machine Learning and Data Mining* (www.ibai-publishing.org/journal/mldm).

Fifteen papers were selected for poster presentations and six for industry paper presentations, which are published in the *ICDM Poster and Industry Proceedings* by *ibai-publishing* (www.ibai-publishing.org).

In conjunction with ICDM, four workshops were run focusing on special hot application-oriented topics in data mining: Data Mining in Marketing (DMM), Data Mining in Life Science (DMLS), the Workshop on Case-Based Reasoning (CBR-MD), and the Workshop Data Mining in Agriculture (DMA). All workshop papers are published in the *workshop proceedings* by *ibai-publishing* (www.ibai-publishing.org).

A tutorial on Data Mining, a tutorial on Case-Based Reasoning, a tutorial on Intelligent Image Interpretation and Computer Vision in Medicine, Biotechnology, Chemistry and Food Industry and a tutorial on Standardization in Immunofluorescence were held before the conference.

We were pleased to give out the the best paper award for the sixth time this year (www.data-mining-forum.de). The final decision was made by the Best Paper Award Committee based on the presentation by the authors and the discussions with the auditorium. The ceremony took place at the end of the conference. This prize is sponsored by *ibai solutions* (www.ibai-solutions.de), one of the leading companies in data mining for marketing, Web mining and e-commerce.

The conference was rounded up by an outlook on new challenging topics in data mining before the Best Paper Award Ceremony.

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

Last, but not least, we wish to thank all the speakers and participants who contributed to the success of the conference. We hope to see you in 2013 in New York at the next World Congress on “The Frontiers in Intelligent Data and Signal Analysis, DSA2013” (www.worldcongressdsa.com) that combines the following three events: the International Conference on Machine Learning and Data Mining MLDM; the Industrial Conference on Data Mining ICDM, and the International Conference on Mass Data Analysis of Signals and Images in Medicine, Biotechnology, Chemistry and Food Industry MDA.

July 2012

Petra Pernert

Organization

Industrial Conference on Data Mining, ICDM 2012

Chair

Petra Perner

IBaI Leipzig, Germany

Committee

Klaus-Peter Adlassnig

Andrea Ahlemeyer-Stubbe

Klaus-Dieter Althoff

Eva Armengol

Brigitte Bartsch-Spörl

Orlando Belo

Isabelle Bichindaritz

Leon Bobrowski

Marc Boullé

Shirley Coleman

Juan M. Corchado

Jeremiah Da Deng

Jeroen de Bruin

Antonio Durado

Peter Funk

Gary F. Holness

Piotr Jedrzejowicz

Janusz Kacprzyk

Mehmed Kantardzic

Mineichi Kudo

David Manzano Macho

Eduardo F. Morales

Stefania Montani

Jerry Oglesby

Wieslaw Paja

Eric Pauwels

Mykola Pechenizkiy

Medical University of Vienna, Austria

ENBIS, The Netherlands

University of Hildesheim, Germany

IIA CSIC, Spain

BSR Consulting GmbH, Germany

University of Minho, Portugal

University of Washington, USA

Bialystok Technical University, Poland

France Télécom, France

University of Newcastle, UK

Universidad de Salamanca, Spain

University of Otago, New Zealand

Medical University of Vienna, Austria

University of Coimbra, Portugal

Mälardalen University, Sweden

Quantum Leap Innovations Inc., USA

Gdynia Maritime University, Poland

Polish Academy of Sciences, Poland

University of Louisville, USA

Hokkaido University, Japan

Ericsson Research Spain, Spain

INAOE, Ciencias Computacionales, Mexico

Università del Piemonte Orientale, Italy

SAS Institute Inc., USA

University of Information Technology and

Management in Rzeszow, Poland

CWI Utrecht, The Netherlands

Eindhoven University of Technology,

The Netherlands

VIII Organization

Georg Ruß

Otto-von-Guericke-Universität Magdeburg,
Germany

Rainer Schmidt

University of Rostock, Germany

Yanbo J. Wang

Information Management Center, China,

Minsheng Banking Corporation Ltd., China

Claus Weihs

University of Dortmund, Germany

Terry Windeatt

University of Surrey, UK

Table of Contents

Data Mining in Medicine and Biology

Application of Classification Algorithms on IDDM Rat Data	1
<i>Rainer Schmidt, Heike Weiss, and Georg Fuellen</i>	
Research Themes in the Case-Based Reasoning in Health Sciences Core Literature	9
<i>Isabelle Bichindaritz</i>	
Research on Application of Data Mining Methods to Diagnosing Gastric Cancer	24
<i>Arnis Kirshners, Serge Parshutin, and Marcis Leja</i>	
SOHAC: Efficient Storage of Tick Data That Supports Search and Analysis	38
<i>Gabor I. Nagy and Krisztian Buza</i>	

Data Mining for Energy Industry

Electricity Consumption Time Series Profiling: A Data Mining Application in Energy Industry	52
<i>Hongyan Liu, Zhiyuan Yao, Tomas Eklund, and Barbro Back</i>	
Wind Turbines Fault Diagnosis Using Ensemble Classifiers	67
<i>Pedro Santos, Luisa F. Villa, Aníbal Reñones, Andrés Bustillo, and Jesús Maudes</i>	

Data Mining in Traffic and Logistic

Bus Bunching Detection by Mining Sequences of Headway Deviations	77
<i>Luís Moreira-Matias, Carlos Ferreira, João Gama, João Mendes-Moreira, and Jorge Freire de Sousa</i>	

Data Mining in Telecommunication

Detecting Abnormal Patterns in Call Graphs Based on the Aggregation of Relevant Vertex Measures	92
<i>Ronnie Alves, Pedro Ferreira, Joel Ribeiro, and Orlando Belo</i>	

Data Mining in Engineering

Real-Time Mass Flow Estimation in Circulating Fluidized Bed 103
Andriy Ivannikov, Mikko Jegoroff, and Tommi Kärkkäinen

Representation in Case-Based Reasoning Applied to Control
 Reconfiguration 113
Ons Lejri and Moncef Tagina

The Influence of Input and Output Measurement Noise on Batch-End
 Quality Prediction with Partial Least Squares 121
*Jef Vanlaer, Pieter Van den Kerkhof, Geert Gins, and
 Jan F.M. Van Impe*

Theory in Data Mining

An Evolving Associative Classifier for Incomplete Database 136
Kaoru Shimada

Improving Classifier Performance by Knowledge-Driven Data
 Preparation 151
Laura Welcker, Stephan Koch, and Frank Dellmann

CWFM: Closed Contingency Weighted Frequent Itemsets Mining 166
*Eunkyong Park, Younghee Kim, Ieejoon Kim, Jaeyeol Yoon,
 Jiyeon Lim, and Ungmo Kim*

Prognostic Modeling with High Dimensional and Censored Data 178
Leon Bobrowski and Tomasz Lukaszuk

Theory in Data Mining: Clustering

SHACUN: Semi-supervised Hierarchical Active Clustering Based on
 Ranking Constraints 194
Eya Ben Ahmed, Ahlem Nabli, and Faïez Gargouri

A Minimum Spanning Tree-Inspired Clustering-Based Outlier Detection
 Technique 209
Xiaochun Wang, Xia Li Wang, and D. Mitch Wilkes

**Theory in Data Mining: Association Rule Mining
 and Decision Rule Mining**

ML-DS: A Novel Deterministic Sampling Algorithm for Association
 Rules Mining 224
*Samir A. Mohamed Elsayed, Sanguthevar Rajasekaran, and
 Reda A. Ammar*

Decision Rules Development Using Set of Generic Operations Approach	236
<i>Wiesław Paja and Mariusz Wrzesień</i>	
Text Mining	
Redundant Dictionary Spaces as a General Concept for the Analysis of Non-vectorial Data	243
<i>Sebastian Klenk, Jürgen Dippon, Andre Burkowski, and Gunther Heidemann</i>	
Human-Centered Text Mining: A New Software System	258
<i>Jonas Poelmans, Paul Elzinga, Alexei A. Neznanov, Guido Dedene, Stijn Viaene, and Sergei O. Kuznetsov</i>	
Text Mining Scientific Papers: A Survey on FCA-Based Information Retrieval Research	273
<i>Jonas Poelmans, Dmitry I. Ignatov, Stijn Viaene, Guido Dedene, and Sergei O. Kuznetsov</i>	
Author Index	289

Application of Classification Algorithms on IDDM Rat Data

Rainer Schmidt¹, Heike Weiss², and Georg Fuellen¹

¹ Institute for Biostatistics and Informatics in Medicine and Aging Research

² Institute for Medical Biochemistry and Molecular Biology,
University of Rostock, Germany

{rainer.schmidt, heike.weiss, georg.fuellen}@uni-rostock.de

Abstract. In our study, we intend to investigate the mechanism of tolerance induction by the modulatory anti CD4 monoclonal antibody RIB 5/2 in insulin dependent diabetes mellitus rats. The aim of this investigation is to identify the key mechanisms of immune tolerance on the level of T cell, cytokine, and chemokine biomarkers in the blood, lymphatic organs, and pancreas. Additionally, it should be possible to define good biomarkers of autoimmunity and tolerance for prediction of diabetes onset. We mainly applied decision trees and later on some other classification algorithms on a rather small data set. Unfortunately, the results are not significant but are good enough to satisfy our biological partners.

Keywords: Insulin dependent diabetes mellitus, Bioinformatics, Machine Learning.

1 Introduction

Type 1 diabetes is an autoimmune disease in which beta cells are exclusively destroyed by the interaction of antigen presenting cells, T cells, and environmental triggers such as nutrients and viral infection [1, 2].

There are two major challenges for prediction and diagnosis of this disease. First, though the analysis of various beta cell autoantibodies and beta cell specific T cells allows a good risk assessment for the progression of autoimmunity, biomarkers related to mechanisms of T cell mediated beta cell destruction and induction of self-tolerance are missing. Second, intervention strategies to block beta cell autoimmunity are not fully understood.

The IDDM (insulin dependent diabetes mellitus) rat is an animal model of spontaneous autoimmune diabetes which is characterized by a fulminant T cell mediated beta cell destruction leading to a full diabetic syndrome in 60 % of the animals around day 60. The narrow time range of islet infiltration between day 40 and day 50 makes this model a valuable tool to study strategies and mechanisms for induction of immune tolerance. Induction of immune tolerance is a promising approach to halt autoimmunity in type 1 diabetes. Anti CD3 antibodies and vaccination with modified

beta cell antigens such as insulin, GAD65, and hsp60 could block autoimmunity and induce self-tolerance in animal models of autoimmune diabetes [3].

These strategies, however, still show limitations that hamper translation into routine clinical use. First, the mechanisms of T cell modulation are still unclear in particular for transition from temporary immune suppression to induction of permanent self-tolerance. Second, despite development of humanized and aglycosylated anti CD3 antibodies the side effects remain severe and raise ethical concerns for treatment of young type 1 diabetes patients.

The intention of our project is

- To elucidate the mechanisms of the modulating anti CD4 antibody RIB5/2 on prevention of autoimmune destruction of beta cells in the IDDM rat model.
- To analyse immune cell (bio-) markers in peripheral blood during progression of autoimmunity and/or induction of self-tolerance.

2 Background and Research Status

Beta cell destruction in type 1 diabetes is a complex process comprising a network between beta cells, antigen presenting cells, autoaggressive T cells, and environmental triggers. Beta cells that are under assault are not passive bystanders, but actively participate in their own destruction process [4, 5]. Overall, many of the cytokine- and virus-induced effects involved in inhibition of beta cell function and survival are regulated at the transcriptional and posttranscriptional/translational level [6]. T-cells modulate the autoimmune process and autoreactive T-cells can transfer diseases [7]. Thus, immune intervention during the prodromal phase or at the onset of overt diabetes will affect the balance between autoreactive and regulatory T cells. Currently it is possible to identify β -cell-specific autoreactive T-cells using standard in vitro proliferation and tetramer assays, but these cell types could also be detected in healthy individuals [8]. Although the analysis of autoantibodies allows an assessment of risk for type 1 diabetes, it is still impossible to draw conclusions about T cell function in the local lymphatic compartment of the pancreas. Notably, there is an extensive knowledge upon activation of T cells and upon induction of self-tolerance on the molecular level of gene expression biomarkers. We hypothesize that biomarkers must be analyzed in a dynamic manner because they shall have specific predictive values for development of autoimmunity at different stages of autoimmunity.

The analysis of gene expression patterns might help to distinguish between T1DM affected subjects and healthy animals at an early stage. In a first experiment, we could demonstrate that analysis of selected genes of T cell differentiation, T cell function, and cytokine expression in whole blood cells at an early prediabetic stage (after 45 days of live), the RT6 T cell proliferation gene was most decisive for diabetes onset in the IDDM rat followed by selectin and neuropilin at the stage of islet infiltration (after 50 days), and IL-4 during progression of beta cell destruction (after 55 days).

3 Data

Several experiments were performed and statistically evaluated. In one of them, for example, it could be shown that the treatment of prediabetic IDDM rats with antibody RIB 5/2 significantly reduces diabetes incidence (from 60% to 11%, see figure 1).

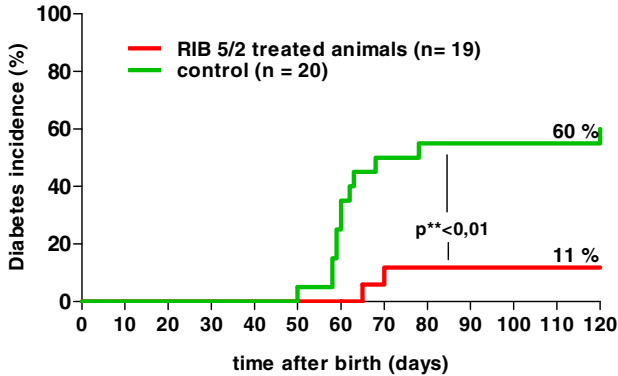


Fig. 1. Protective effect of RIB/2 CD4 modulation on diabetes incidence and age of manifestation ($p < 0.01$, chi-test)

For recent experiments, data from twelve rats were available. They were monitored for gene expression data in blood immune cells for functional gene clusters on the days 30, 35, 40, 45, 50, 55, 60, 65, 70, 80, and 90 of their life. However, just the days between 45 and 60 are assumed to be important of the prediction whether a rat will develop diabetes or not. Six of the twelve rats developed diabetes, three did not, and another three rats (background strain) were diabetes resistant because of the way they had been bred. Unfortunately, due to problems of the measurement facilities the data quality is rather poor. Many data are missing and some are obviously incorrect, especially for the early and the late measurement time points. However, as mentioned above, the most important measurement time points are in the middle. So, for some measurement time points, data from just eleven of the twelve rats were used.

4 Experimental Results

In the experiments data of the following measurement time points were used: 45, 50, 55, and 60 days of life. The attributes are eighteen preselected genes and biomarkers. The class labels are “diabetes”, “no diabetes”, and “background strain”.

Since we wanted to get attributes that are most decisive for the classification, we applied decision trees, which do not just provide the most decisive attributes but also their decisive values. The C4.5 decision tree algorithm, which was originally developed by Ross Quinlan [9], was applied in form of its J48 implementation in the

WEKA environment [10]. Later on, we also applied other classification algorithms that are provided in WEKA, like “random forest”, for example.

The tree for day 50 is depicted in figure 2 and states the following. If the gene expression value of selectin is bigger than 2.14 a rat probably belongs to the background strain, otherwise if the gene expression value of neuropilin is bigger than 0.63 a rat probably does not develop diabetes, otherwise it probably develops diabetes.

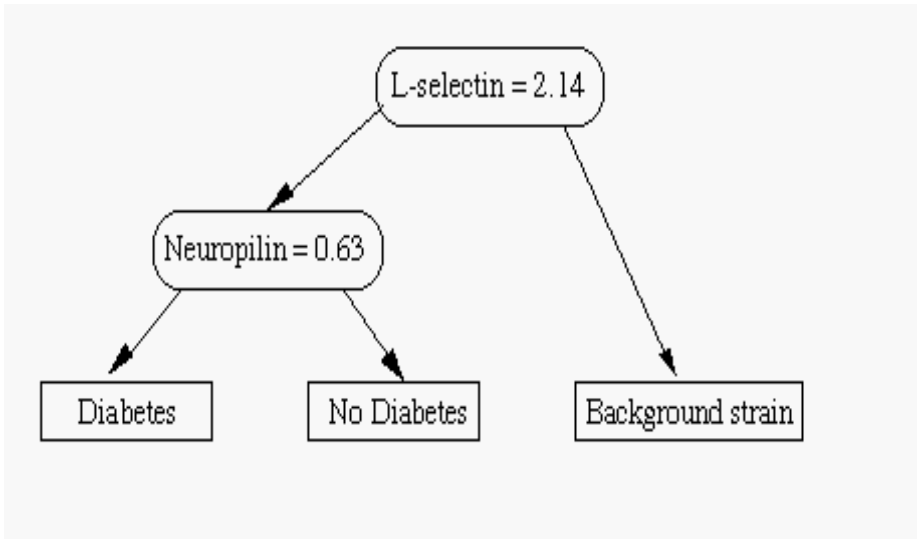


Fig. 2. Decision tree for day 50

The results for the days 45, 50, and 55 are depicted in figure 3. There are three trees depicted, the left one is for day 45, the right one is for day 55. The tree for day 50 (the same as in figure 2) is depicted in the middle.

At the beginning of infiltration (day 45) the RT6 gene expression, responsible for the correct thymic development of T-cells, may decide whether autoimmunity could develop. At a stage of islet infiltration (day 50) selectin and neuropilin gene expression decides whether primed T-cells will infiltrate the endocrine pancreas for beta cell destruction. During progression of beta cell destruction (day 55) IL-4 as a T cell stimulating cytokine is crucial for the progression of beta cell infiltration.

5 Validation

For the IDDM rat model we have started to calculate relative risk coefficients for development for diabetes. Though the data set is very small, the biologists that are involved in the project are very happy with the results and can explain them (caption of figure 1). However, because of the extremely small data set (twelve rats), the set was not split into a learning and a test set. The trees are computed on the training set.

So, next Information Gain [10] was considered, on which decision trees are based. WEKA provides them as “attribute selection”. Usually, the values are between 0 and 1. In three of four trees the decision was obvious. For day 45, for example, the value of *rt6a* is 0.811, whereas the values of all other attributes are 0. Just, for day 50 the decision is obvious but the whole situation is not completely clear, because the Information Gain values are 0.959 for *l-selection* and 0.593 for *il-4* and for *neuropilin*. Furthermore, in the tree *neuropilin* is used to separate between “diabetes” and “no diabetes”. So, the background strain was excluded and Information Gain was used just to classify “diabetes” and “no diabetes”, with the result that *neuropilin* was the first choice.

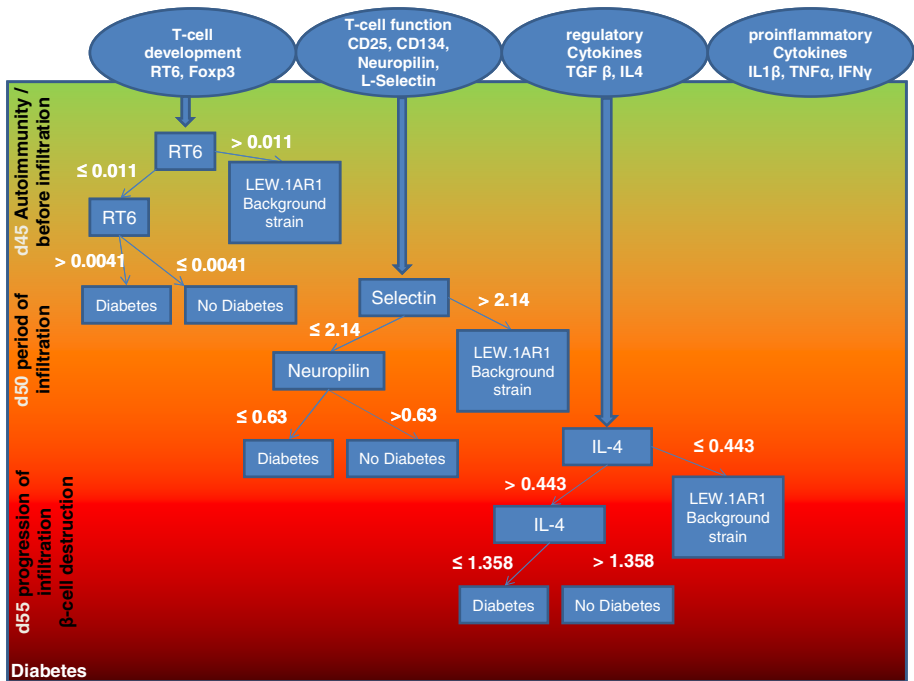


Fig. 3. Relative gene expression levels analysed with the C4.5 algorithm. The numbers heading the arrows indicate the threshold values of gene expression normalized to GAPDH quantified by real-time RT-PCR analysis.

Afterwards some standard classification methods provided by WEKA were applied (naïve bayes [11], nearest neighbor [12], random forest [13], J48, and support vector machines [14]). Except for the decision tree algorithm J48 these methods show just the classification results but they do not show which attributes have been used for the classification. In table 1 results are shown just for day 50 as an example. An inner cross validation is provided by WEKA. Because of the small size of the data set 3-fold cross validation was applied instead of the usual 10-fold cross validation. First,

the classification algorithms were applied on the whole data sets (table 1) and secondly on the data sets without background strain (table 2). However, the differences are very small.

Table 1. Accuracy and Area Under the Curve for day 50 for the complete data set

Method	Accuracy (%)	AUC
Naïve Bayes	58.3	0.52
Nearest Neighbor	75	0.75
Random Forest	66.7	0.80
J48	66.7	0.76
SVM	58.3	0.65

Table 2. Accuracy and Area Under the Curve for day 50 for the data set without background strain

Method	Accuracy (%)	AUC
Naïve Bayes	55.6	0.42
Nearest Neighbor	77.8	0.67
Random Forest	66.7	0.53
J48	66.7	0.61
SVM	55.6	0.50

6 Discussion

The application of Machine Learning (classification) methods has become popular in bioinformatics. This is already reflected at the ICDM conferences (e.g. [15,16]). In our application, the analysis of gene expression patterns might help to distinguish between T1DM affected subjects and healthy animals at an early stage. In a first experiment, we could demonstrate that analysis of selected genes of T cell differentiation, T cell function, and cytokine expression in whole blood cells at an early prediabetic stage (after 45 days of live), the RT6 T cell proliferation gene was most decisive for diabetes onset in the IDDM rat followed by selectin and neuropilin at the stage of islet infiltration (after 50 days), and IL-4 during progression of beta cell destruction (after 55 days).

However, so far the data set is very small and, probably because of poor data quality, the cross-validated classification results are not significant (see tables 1 and 2). Nevertheless, the generated decision trees perform well, certainly just on the training set, but nearly all of them can be very well explained by the biochemical experts.

So, because of the small size of the data set, we tried to breed some more specific rats. Unfortunately, because of a virus in the rat laboratory this was just partly successful. Furthermore, since the data quality was poor, we applied another measurement facility. We got a better data quality for just eight new rats. However,

both data sets should not be joined together. Because of different measurement facilities they are not compatible with each other.

The eight new cases belong to just two classes. Six rats developed diabetes, the other two ones did not. With such a data set the application of decision trees does not seem to be reasonable, because there is a big chance that one attribute can be found that might be sufficient to split away the two rats from the remaining six diabetes rats.

Actually, the situation is even worse. Most attributes can be used to distinguish between the two classes. Since all attributes we found in our earlier experiments (see figure 3) are among them, this new data set supports our findings. However, this support is rather weak, because for these new data many alternative attributes can also be used to separate between the two classes.

Unfortunately, the breeding and the data collection of these specific rats is expensive and time consuming. Furthermore, sometimes the breeding may even fail (see above).

However, our results (especially the decision trees) are not just good enough to satisfy our biological partners but also to get the funding, so that we are going to start another breeding attempt.

References

1. Akerblom, H.K., Vaarala, O., Hyoty, H., Ilonen, J., Knip, M.: Environmental factors in the etiology of type 1 diabetes. *Am. J. Med. Genet.* 115, 18–29 (2002)
2. Jun, H.S., Yoon, J.W.: A new look at viruses in type 1 diabetes. *Diabetes Metab. Res. Rev.* 19, 8–31 (2003)
3. Ludvigsson, J., Faresjo, M., Hjorth, M., et al.: GAD treatment and insulin secretion in recent-onset type 1 diabetes. *N. Engl. J. Med.* 359, 1909–1920 (2008)
4. D’Hertog, W., Overbergh, L., Lage, K., et al.: Proteomics analysis of cytokine-induced dysfunction and death in insulin-producing INS-1E cells: new insights into the pathways involved. *Mol. Cell Proteomics* 6(21), 80–99 (2007)
5. Rasschaert, J., Liu, D., Kutlu, B., Cardozo, A.K., Kruhoffer, M., ØRntoft, T.F., Eizirik, D.L.: Global profiling of double stranded RNA- and IFN-gamma-induced genes in rat pancreatic beta cells. *Diabetologia* 46, 1641–1657 (2003)
6. Gysemans, C., Callewaert, H., Overbergh, L., Mathieu, C.: Cytokine signalling in the beta-cell: a dual role for IFNgamma. *Biochem. Soc. Trans.* 36, 328–333 (2008)
7. Lampeter, E.F., McCann, S.R., Kolb, H.: Transfer of diabetes type 1 by bone-marrow transplantation. *Lancet* 351, 568–569 (1998)
8. Schloot, N.C., Roep, B.O., Wegmann, D.R., Yu, L., Wang, T.B., Eisenbarth, G.S.: T-cell reactivity to GAD65 peptide sequences shared with coxsackie virus protein in recent-onset IDDM, post-onset IDDM patients and control subjects. *Diabetologia* 40, 332–338 (1997)
9. Quinlan, J.R.: *C4.5 Programs for Machine Learning*. Morgan Kaufmann, San Mateo (1993)
10. Hall, M., et al.: The WEKA data mining software: An update. *SIGKDD Explorations* 11(1), 10–18 (2009)
11. Gan, Z., Chow, T.W., Huang, D.: Effective Gene Selection Method Using Bayesian Discriminant Based Criterion and Genetic Algorithms. *Journal of Signal Processing Systems* 50, 293–304 (2008)

12. Cost, S., Salzberg, S.: A weighted Nearest Neighbor Algorithm for Learning with Symbolic Features. *Machine Learning* 10(1), 57–78 (1993)
13. Breiman, L.: Random Forest. *Machine Learning* 45(1), 5–32 (2001)
14. Platt, J.: *Avances in Large Margin Classifiers*, pp. 61–74. MIT-Press (1999)
15. Bichindaritz, I.: Methods in Case-Based Classification in Bioinformatics: Lessons Learned. In: Perner, P. (ed.) *ICDM 2011*. LNCS, vol. 6870, pp. 300–313. Springer, Heidelberg (2011)
16. Perner, J., Zotenko, E.: Characterizing Cell Types through Differentially Expressed Gene Clusters Using a Model-Based Approach. In: Perner, P. (ed.) *ICDM 2011*. LNCS, vol. 6870, pp. 106–120. Springer, Heidelberg (2011)

Research Themes in the Case-Based Reasoning in Health Sciences Core Literature

Isabelle Bichindaritz

State University of New York
Oswego, New York 13126, USA
ibichind@gmail.com

Abstract. Research in case-based reasoning (CBR) in the health sciences started more than 20 years ago and has been steadily expanding during these years. This paper describes the state of the research through an analysis of its mainstream, or core, literature. The methodology followed involves first the definition of a classification and indexing scheme for this research area using a tiered approach to paper categorization based on application domain, purpose of the research, memory organization, reasoning characteristics, and system design. A research theme can be tied to any of the previous classification elements. The paper further analyzes the evolution of the literature, its characteristics in terms of highest impact, or most cited, papers, and draws conclusions from this analysis. Finally, a comparison with the themes automatically learned through clustering co-citations matrices with the Ensemble Non-negative Matrix Factorization (NMF) algorithm in the CBR conference literature is proposed. This comparison helps better understand the main characteristics of the field and propose future directions.

Keywords: case-based reasoning, classification, biomedical informatics, biometrics, text mining.

1 Introduction

The field of Case-Based Reasoning (CBR) in the Health Sciences (CBR-HS) [1] has seen a tremendous growth in the last decade. An international group of researchers performs its research mainly in this domain, and constitutes the core CBR-HS research community. Seven specialized conference workshops have been held consecutively between 2003 and 2009 focused solely on this topic. In addition, six journal special issues on CBR-HS were published in the journals *Artificial Intelligence in Medicine* [2][3][4], *Computational Intelligence* [5][6], and *Applied Intelligence* [7]. The domain has been the subject of several survey papers as well, mostly qualitative in nature, hence the need to track the evolution in the research in a more systematic and automatic manner. We developed a classification and indexing scheme for CBR research in the Health Sciences to make possible the meta-analysis of this interdisciplinary research area [1] in a semi-automatic manner. This paper details knowledge of CBR-HS gained by building and using this classification scheme and the research

trends identified in terms of application domains, application purposes, system memory, reasoning, and design, as well as evolution of number of papers, citations, and research themes. In addition, a comparison is proposed with an automatic clustering method called Ensemble Non-negative Matrix Factorization (NMF) [8] to determine how the major themes in the CBR conference literature differ from those in the core CBR-HS literature.

2 Methods

The specific application of CBR to the health sciences has been discussed in several surveys [9, 10, 11, 12, 13, 14, 15]. However recent trend analyzes in CBR as a whole failed to identify CBR-HS as a sub-research area through automatic methods [8]. This may in particular be due to the variety of application domains comprising the health sciences, which prompts for the need to index systems capable in particular of grouping documents related to, for example, oncology, diabetology, phrenology and so forth. Therefore we developed a classification and indexing system capable of drilling down and rolling up in its different components and presented in detail elsewhere [1]. This domain-specific indexing is enabled by the use of one of the most used classification schemes in the health sciences: the Medical Subject Headings (MeSH) [16]. Like most other classifications, it uses a tree like structure where broader categories are narrowed down with each branch and branches are represented by dots.

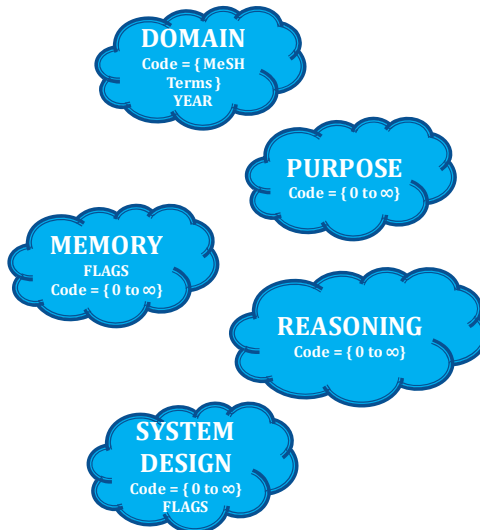


Fig. 1. CBR Health Sciences tiered classification scheme

The papers selected for CBR-HS in this paper cover all the 16 EWCBR (European Workshop on Case Based Reasoning), ECCBR (European Conference on Case Based Reasoning), and ICCBR (International Conference on Case Based Reasoning) conferences from 1993 until 2011, the 7 Workshops on CBR in Health Sciences, the 5

special issues on CBR in the Health Sciences, the 2 DARPA workshops of 1989 and 1991, which preceded the CBR official conferences, and the survey papers on CBR in the Health Sciences. We also added papers preceding the papers published in the official CBR-HS venues, before they existed. These papers were identified by the group of CBR-HS researchers who prepared the 2007 survey. 156 papers were indexed with the CBR-HS classification scheme, presented in the next section. Therefore the terminology learned for the classification has been refined on these 156 papers.

3 Classification System

Figure 1 presents the tiered architecture of the CBR-HS classification scheme. There are five distinct categories (domain, purpose, memory and case management, reasoning, and system design) defined in this section. A research theme can be selected by researchers among any of these categories, to characterize the main research hypothesis and findings of the paper. Codes have been created to represent each classification category. We refer the reader to another article [1] for the coding details.

1. **Domain:** The range of domains, such as for example oncology or diabetology, in the health sciences fields is vast and, as a result, it was chosen as the first level of classification. However, rather than creating a new set of descriptors, it is proposed to use the MeSH descriptors [16], of which there are over 24,000 that cover just about every aspect of the health sciences. Along with the domain, another primary means of discriminating the relevance of an article is its publication date.
2. **Purpose:** The purposes, or tasks, of CBR systems have been thoroughly discussed in many articles summarizing the CBR-HS domain. One of the first papers to survey the field in 1998, by Gierl et al., used the purpose as the primary means to subdivide the different systems [9]. In their paper, Gierl et al. specified four main purposes: diagnosis, classification, planning, and tutoring. Later, both Holt et al. 2006 [13] and Nilsson and Sollenborn 2004 [12] used the same four descriptors. In the early years the majority of systems were diagnostic in nature, but in recent years more therapeutic and treatment systems have been developed [14]. We have replaced planning by treatment since most of the time planning refers to treatment planning. However, planning tasks may involve not only treatment but also other aspects such as diagnosis assessment, which often consists in a series of exams and labs orchestrated in a plan. Planning is a classical major task performed by artificial intelligence systems. Therefore planning is listed in our system as a design option and thus can be added to the treatment choice in the purpose dimension. CBR systems generally support either medical clinical work, or research. Therefore we have added these as top level purpose categories (see Table 2). In the clinic, decision support systems support mostly diagnosis, treatment, prognosis, follow-up, and/or classification, such as in image interpretation. More recent articles require to differentiate between the purpose of the system developed, which is generally a clinical purpose, from the

- purpose of the research paper, which can be, among others, a survey paper or a classification paper like this one. Some papers focus on formalization, a method, or a concept. Among these, the evaluation of a system can be performed more or less thoroughly. This is an important dimension to note about a research paper: whether the system was tested only at the system level, which is the most frequent, at the pilot testing level, at the clinical trial level, or finally whether the system is in routine clinical use.
3. **Memory and case management:** This is a very broad category and could easily be subdivided. It encompasses both how the cases are represented and also how they are organized in memory for retrieval purposes and more (see Table 3). As a result, it is made up of more than one code. The first part of the code represents the format of the cases. The primary types being images, signals, mass spectrometry, microarray, time series data and regular attribute/values pairs, which is used by the majority of the systems. Similar to the different formats of data are the flags that represent what kinds of memory structures the CBR system uses to represent the data, such as ground cases (G), prototypical cases (P), clusters (L), or concepts (O). Lastly, when it comes to memory management there are potentially an infinite number of possibilities, some of which may never have been used before. The main types, however, represent how the memory is organized, whether it is flat or hierarchical, what kind of hierarchical structure, such as decision tree, concept lattice, conceptual clustering tree, or others.
 4. **Reasoning:** This category regroups the inferential aspects of the CBR. Classically, retrieve, reuse, revise, and retain have been described. Nevertheless, researchers have often added many more aspects to the inferences, such that it is best to keep this category open to important variations. Each of these parts of the reasoning cycle can be hierarchically refined so that a tree is formed here also.

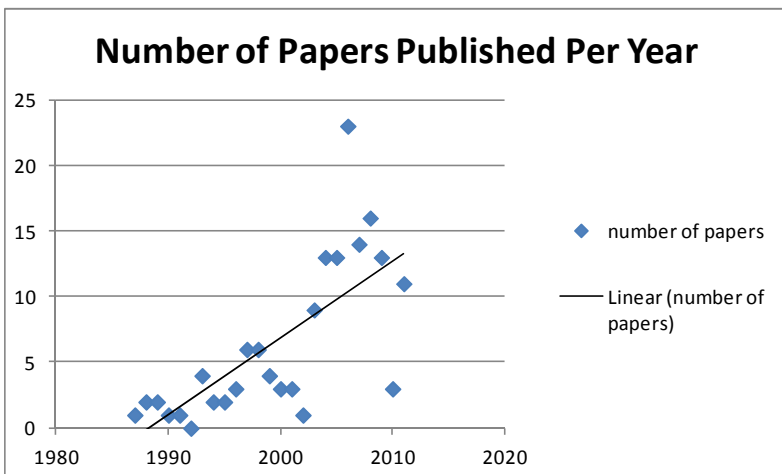


Fig. 2. Evolution of the number of papers published

5. System design: The construction of the CBR system specifies what technologies it uses. This area of classification may not seem intuitive at first, but upon the examination of CBR systems it can be seen that many use a combination of technologies, not just case-based reasoning. The most common technology used in conjunction with CBR is rule-based reasoning; however some systems combine CBR with information retrieval, data mining, or other artificial intelligence methods. See table 4 for an example of different possible construction classifications. If the construction of the system does use additional technologies, a flag should be appended to the end of the code to denote whether the case-based reasoning is executed separately. Also, an additional flag is used to designate CBR's role in the system, whether primary, secondary, or equivalent.

Table 1. A ranked list of the top 10 highest impact papers in the core CBR-HS collection based on total citation count

#	Paper	Year	Citations
1	Concept learning and heuristic classification in weak-theory domains Porter, Bareiss & Holte [17]	1990	290
2	Reasoning about evidence in causal explanations Koton [18]	1988	239
3	Protos: an exemplar-based learning apprentice Bareiss, Perter & Wier [19]	1988	156
4	Case-based reasoning in CARE-PARTNER: gathering evidence for evidence-based medical practice Bichindaritz, Kansu & Sullivan [20]	1998	86
5	Cased-based reasoning for medical knowledge-based systems Schmidt, Montani, Bellazzi, Portinale & Gierl [10]	2001	83
6	Using experience in clinical problem solving: introduction and Framework Kolodner & Kolodner [21]	1987	81
7	A two layer case-based reasoning architecture for medical image understanding Grimnes & Aamodt [22]	1996	76
8	Case-based reasoning in the health sciences: what's next? Bichindaritz & Marling [11]	2006	72
9	An architecture for a CBR image segmentation system Perner [23]	1999	69
10	Advancements and trends in medical case based reasoning: an overview of systems and system development Nilsson & Sollenborn [12]	2004	65

4 Global Picture

The global picture of the core CBR-HS literature shows a total of 156 papers being published between 1987 and 2011 from 179 different authors from all over the world. The average number of papers per author is 2.27, and the range is 1 to 27. We searched these papers in Google Scholar to count their number of citations and calculated a total of 3237 citations.

The evolution of the number of papers is provided on Fig. 1. It shows a regular increase in the research productivity in this domain, which attests of the vitality of the field. This graph demonstrates in particular that the number of papers by year has seen a rapid increase after 2003 – corresponding to the first workshop dedicated to CBR-HS (see Figure 1).

In terms of impact, Table 1 lists the 10 highest impact papers based on their number of citations in Google Scholar, after removing the number of self-citations (only the order of the papers changes if taking into account all citations). It is interesting to note that the pioneering papers in the domain are ranked in positions #1, 2, 3, and 6. These papers preceded the creation of the CBR-HS research field, however have impacted the field tremendously. These papers do not refer to the term of CBR yet, however they have served to define the feasibility and direction of this research. In that sense, they can be regarded as its seed papers. The other papers took about 10 years to emerge from the tracks defined by the seed papers (paper #4 in particular). Paper #5 is the first survey paper in CBR-HS, and papers #8 and 10 are later surveys. Papers #7 and 9 represent the seeds in a group of CBR-HS papers devoted to the research theme of medical image interpretation.

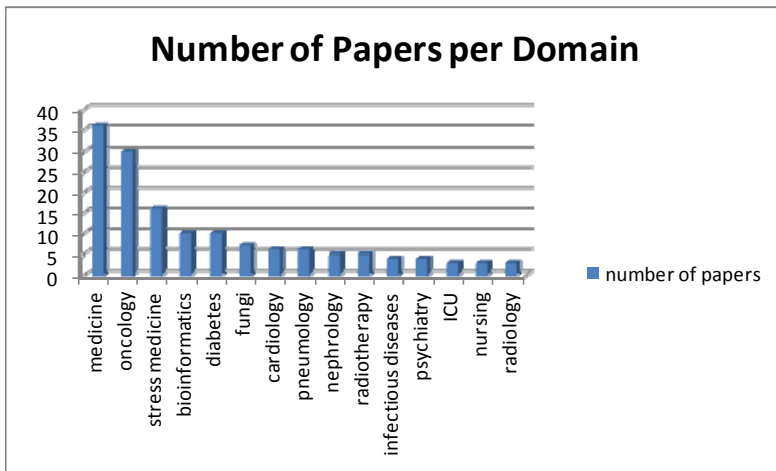


Fig. 3. Domains the most studied by CBR-HS papers

5 Analysis of Classes

Based on the classification we have defined, interesting research themes appear in terms of domain, purpose, memory, reasoning, and design. Since most papers focus

both on a domain and another dimension such as design for example, the papers contribute to several classes. In addition, in each class, they also very often contribute to several categories, such as treatment and diagnosis for example.

Domains

The 156 papers cover 38 domains all together. Although the domains of application all belong to the Health Sciences, some domains are more represented than the other ones. The most represented domain is medicine with 36 papers as a whole, which correspond to either survey papers, editorials, or general frameworks and concepts applicable to any health sciences domain. Close second comes oncology (30 papers), then further come stress medicine (16 papers), diabetes (10 papers), fungi detection (7 papers) – which could have been added to the infectious diseases papers, cardiology and pulmonology (6 papers each), nephrology and radiotherapy (5 papers each), infectious diseases and psychiatry (4 papers each), and intensive care (ICU), nursing, and radiology (3 papers each). All the other domains count less than 3 papers. It is interesting to note in particular that cancer, being a very prominent disease, is studied by several CBR-HS teams in the world.

Table 2. Main Purpose Themes and the corresponding number of papers

Purpose	#	Purpose	#
Medical Purpose	156	CBR-HS Research Purpose	32
Decision Support	136	Survey	17
Treatment/therapy	46	Evaluation / testing	8
Diagnosis	36	Role of CBR	4
Classification	27	Concept	2
Interpretation	13	Formalization	1
Prognosis / prediction	7		
Follow-up	5		
Assessment	2		
Medical research support	8		
Quality control / monitoring	3		
Information retrieval / navigation	3		
Tutoring	2		
Parameter configuration	2		
Drug design	1		
Explanation	1		

Purpose

Among the 24 purposes listed in these papers, we can distinguish between medical purpose, tied to the application domain, and research purpose, tied to the CBR-HS domain.

In terms of medical purpose, 46 papers propose treatments / therapies (among which two propose prescriptions), 36 propose diagnosis recommendations, 27 classifications, 17 papers refer globally to decision support (to which we can add the

sub-types of decision-support tasks – see Table 2), and there are additional decision-support tasks such as interpretation (mostly for image interpretation). Other papers propose to help medical research (8), quality control and monitoring (3), and information retrieval or case-base navigation (3), among several other medical purposes.

In terms of research or methodological purpose, 17 papers are survey, editorial, or systematization papers, 8 papers focus on evaluation methods, 4 papers investigate the role CBR can play in medical domains, and a few papers focus on formalization, concepts, and methods.

It is notable that 28 papers describe several purposes, for example they tackle both diagnosis and treatment decision-support, although each of these tasks alone, given its complexity, could be the topic of an entire paper. Another important characteristic is that several systems focus on differential diagnosis, which involves the value of diversity in the diagnostic recommendation.

Table 3. Sample Memory and Case Management Themes

Generalized Memory Structures	#	Data Types	#
Prototypes	27	Time Series / signals / sensor data	24
Clusters	4	Images	17
Categories	3	Microarray data / genetic sequences	10
Generalized cases	3	Text	7
Inverted indexes	2	Scenarios	2
Schemas	2	Graphs	1
Scenarios	2	Networks	1
Concepts	1	Plans	1
Trends	1	Visio-spatial data	1

Memory and Case Management

Memory structures and organization refers to at least 24 different concepts, which encompass generalized memory structures and a variety of ground cases which can be identified by their case data types (see Table 3). In addition to traditional ground cases or exemplars, which appear in almost all papers, the most represented memory structures are prototypes (27 papers), closely followed by time series ground cases (24 papers, most of them being from signals). Further come image ground cases (17 papers), microarray data ground cases (10 papers), text ground cases (7 papers), clusters (4 papers), categories (3 papers), generalized cases (3), inverted indexes (2 papers), scenarios (2 papers), and schemas (2 papers). The other listed memory structures contain, among others, networks, graphs, multimedia data, plans, structured cases, and visio-spatial cases.

In terms of memory organization, the various types are exemplified in these systems, ranging from flat memories, to decision trees and concept hierarchies. Hierarchical organizations are very prominent in the systems using the generalized memory structures (there are 45 of these papers).

Table 4. System Design Classification

Artificial intelligence methodology / component	#	Biomedical methodology / component	#
Machine learning & data mining	34	Clinical guidelines	6
Prototype learning / generalized case learning	11	Electronic medical records	3
Feature mining / key sequence learning	6		
kNN	5		
Statistical learning	4		
Conceptual clustering	3		
Text mining / case mining	3		
Genetic algorithms	2		
Feature selection / dimensionality reduction	7		
Rule based reasoning	16		
Temporal abstraction	14		
Fuzzy logic	9		
Information retrieval	9		
Knowledge discovery	7		
Knowledge-based systems / semantic Web	6		
Planning	6		
Knowledge acquisition	5		
Temporal reasoning	3		

Reasoning

In CBR-HS the vast majority of systems refer to retrieval and similarity assessment (92 papers) as well as another form of reasoning. Next, maintenance is also well represented (16 papers), as well as adaptation and reuse (15 papers). Further are represented: retain step (8 papers), indexing (5 papers), and revision (5 papers). The retain step could be combined with case base maintenance, even though authors using one term sometimes do not use the other term. Most systems perform several reasoning steps, even though the papers studied did not detail these steps, focusing on retrieval aspects instead. Many papers deal with several reasoning steps in the same paper.

System Design

Main characteristics of developed systems describe the types of components involved in building CBR systems in the health sciences. Although there are many “pure” CBR systems, most systems describe a combination of components to manage to solve a problem in the application domain, thus making them hybrid systems. The role of CBR in the hybrid system is most of the time the primary methodology, although many systems report methodologies of equivalent role. Few describe CBR as a

secondary methodology. There are mostly two types of hybrid methodologies: those coming from artificial intelligence and more broadly computer science (such as ambient systems), and those coming from the field of biomedical informatics.

We have listed 52 different methodologies added to CBR (see Table 4). The main methodologies are: machine learning and data mining (34 papers), with different methods such as prototype and generalized case learning, feature mining, kNN, conceptual clustering, statistical learning, text mining, case mining, and genetic algorithms. In second place and beyond come rule-based reasoning (16 papers), temporal abstraction (14 papers), fuzzy logic (9 papers), information retrieval (9 papers), knowledge discovery (7), knowledge-based systems (6), and planning (6) combinations.

However two categories are specific to medical domains: clinical guidelines integration, and electronic medical records integration.

Table 5. Major research themes in the core CBR-HS literature based on the number of papers addressing them

#	CBR-HS core literature	Number of papers
1	Reasoning: retrieval & similarity assessment	92
2	Purpose: treatment or therapy decision-support	46
3	Purpose: diagnosis decision-support	36
4	Design: machine learning / data mining combination	34
5	Domain: oncology	30
6	Memory: prototypes	27
7	Purpose: classification	27
8	Memory: time series / signals / sensor data	24
9	Memory: images	17
10	Design: temporal abstraction & reasoning	17
11	Design: rule based reasoning combination	16
12	Reasoning: case base maintenance	16
13	Reasoning: adaptation	15
14	Purpose: interpretation decision-support	13
15	Memory: microarray data / genetic sequences	10
16	Design: prototype learning / generalized case learning	11
17	Design: fuzzy logic combination	9
18	Design: information retrieval combination	9
19	Reasoning: retain	8
20	Purpose: evaluation & testing	7

Top Twenty Research Themes

Combining results from the previous sub-sections, we get a clear picture of the major themes in the core CBR-HS literature (see Table 5). We can also note that some very important themes are not in the top twenty research themes, however they are promising and very important for the future development of the field (clinical guidelines integration and electronic medical records integration are some examples) [11].

A research theme can pertain to any of the classification tiers previously presented. Table 5 ranks the top twenty research themes in term of number of papers dealing with it in some way. Since the papers often cover several of these research themes, the sum of these figures does not have to be equal to the number of papers.

With this simplification, the major research themes are the ones ranked #1 through 8 since there is a clear break in number of papers between 24 and 17, in terms of number of papers. These major research themes are therefore:

- In terms of the reasoning dimension, retrieval and similarity assessment (#1);
- In terms of the purpose dimension, treatment / therapy (#2), diagnosis (#3), and classification (#7) decision-support;
- In terms of the design dimension, machine learning / data mining combination (#4);
- In terms of the domain dimension, oncology (#5);
- In terms of the memory dimension, prototypes (#6) and time series / signals / sensor data (#8).

6 Comparison with CBR Conference Research Themes

Greene et al. have identified through an automatic method, called NMF, a number of major themes in the CBR conference literature [8]. It is interesting to compare the major themes identified above with those they have identified (see Table 6).

Table 6 shows the themes in correspondence, namely Case base maintenance, Case retrieval & similarity assessment, Adaptation, Image analysis, Textual CBR, Creativity & knowledge-intensive CBR, CBR on temporal problems, and Structural cases. A ‘Yes’ in the 3rd column indicates that this theme from the CBR conference literature [8] is also present along the highest ranked themes in the CBR_HS literature as identified in the present paper. The ‘#’ symbol refers to the ranking in either the CBR conference literature [8] (2nd column) or the CBR-HS core literature (4th column).

As for CBR conference literature main research themes not represented on Table 6, they are of interest for suggesting future research themes in CBR-HS:

- Recommender systems & diversity: diversity is an important aspect for differential diagnosis. Even though a few CBR-HS systems show some interest in this direction, it is a promising topic to focus on for the future. The spread of health-related online communities and social networks may very well join the research efforts in recommender systems. In addition, the team-based work in the clinic could also take example on this core CBR research for CBR-HS systems.
- Learning similarity measures: even though CBR-HS systems have not yet applied this to their systems yet, it is probably a potential improvement to test.
- Conversational CBR: very few CBR-HS systems actually interact so closely with healthcare professionals, however this could become very

important for patient-centered CBR-HS systems (another potentially very important research direction).

- Feature weighting and similarity: CBR-HS systems in bioinformatics have started researching in this direction, which is connected with the feature mining themes and the feature selection / dimensionality reduction theme. However it is not clear from the Greene et al. paper whether they encompass these in this category [8] – it would make sense to connect them.
- Games & chess: the field of serious games will provide in the future opportunities for common projects with CBR-HS, for example for pain management and phobia treatment.
- Scheduling & agents: there are potential common research projects in health care management and in public health.

Table 6. Comparison of research themes between the CBR conference literature and the CBR-HS core literature (the ‘#’ columns represent the ranking in the articles of reference)

CBR conference literature	#	CBR-HS core literature	#
Recommender systems & diversity	1	Not a major theme currently	N/A
Case base maintenance	2	Yes	12
Case retrieval & similarity assessment	3	Yes	1
Learning similarity measures	4	Not a major theme currently	N/A
Adaptation	5	Yes	13
Image analysis	6	Yes	9
Textual CBR	7	Yes	18
Conversational CBR	8	Not a major theme currently	N/A
Feature weighting & similarity	9	Not a major theme currently	N/A
Creativity & knowledge-intensive CBR	10	Yes – rule-based combination	11
CBR on temporal problems	11	Yes	8
Games & chess	12	Not a major theme currently	N/A
Scheduling & agents	13	Not a major theme currently	N/A
Structural cases	14	Yes – prototypes & prototype learning / generalized case learning	6

In terms of research themes little represented in the CBR conference literature, we can list those with a medical purpose, in particular treatment / therapy decision-support (#2), diagnosis decision-support (#3), and classification decision-support (#7). Of course the oncology domain (#5) is not a major research them in the CBR conference literature. We can also list complex structured cases and complex case data types in memory, as they exist in biomedical domains, such as prototypes (#6) and time series / signals / sensor data (#8). Hybrid systems were not identified either as a major theme in the CBR conference literature, hence the non-existence of machine learning / data mining combination (#4) for example.

We would also like to comment on the highly cited papers in CBR-HS, namely papers #1 (PROTOS [17], 290 citations), #2 (CASEY [18], 239 citations), #3 (PROTOS [19], 156 citations), and #6 (SHRINK [21], 81 citations). These papers clearly demonstrate that CBR-HS papers can have an impact as high and even higher as those in the CBR conference literature, where the top ranked papers receive 137, 117, 92, and 82 citations on Google Scholar. None of these CBR-HS papers clearly label themselves as CBR papers, and we may wonder whether this is not in part an explanation for their success. They present their concepts and ideas more in cognitive terms understandable to any researcher in biomedical or artificial intelligence domains, which probably contributes to making them attractive to a broader audience.

It is also interesting to note that the highest ranked CBR-HS paper from the CBR conference papers [20] counts 86 citations after removing self-citations (98 otherwise), which positions it at the top 4th position both in the CBR-HS classification (see Table 1) and potentially in the Greene et al. classification [8]. Therefore CBR-HS papers published at CBR conferences can reach a citation count comparable to that of highly cited non applied papers. This is encouraging for authors publishing mostly in the CBR conferences.

7 Discussion and Future Plans

The CBR-HS classification system is being incrementally built. The different categories and each category's list of descriptors are by no means exhaustive. However it proved useful for indexing and tracking CBR-HS research literature. With its system of tiers, some of which may be omitted, this system is very flexible and can index either fielded applications, frameworks, or survey papers. This study has identified interesting research themes characteristic of applied domains such as health sciences domains. The classification system allows for an easy tracking of these trends over time.

In comparison with previous survey papers, which are more qualitative in nature, the results presented in this paper share many important facts. For example, in the most recent survey, Begum et al. classify the CBR-HS papers between those that are purpose-oriented and those that are construction-oriented [15]. We also classify them in terms of their purpose dimension and in terms of their design dimension. The major themes they list correspond to a large extent to the ones we have identified; however we quantify the weight of each group of papers. In addition we have conducted a more exhaustive study on a larger pool of papers (156) and along more dimensions, made possible by the indexing simplification provided by the classification system. We intend to continue tracking progress in CBR-HS through this indexing mechanism and to make the papers and their indexing available from a Web-site to better showcase accomplishments in CBR-HS.

Our next goal is to attempt an automatic classification with NMF algorithm as described by Green et al. [8]. Although we do not expect very interesting results from this additional study, since these authors report that they could not identify a cluster for the CBR-HS domain, it is possible that some sub-clusters could overlap with the

ones we found with a semi-automatic indexing of the papers, as presented in this document. These automatically found clusters may also suggest some indexing terms and concepts we may have overlooked in the current study. In addition, the co-citation analysis will provide a different view of the most influential literature, however, as shown in Greene et al., the overlap with the number of citations is expected to be very important [8].

Another planned activity is to provide an automatic or semi-automatic indexing of the articles. Right now, the indexing is humanly made, however we are in the process of attempting to index the papers largely automatically – under the supervision of an expert, which is how current literature indexing is accomplished on a large scale. A completely automatic indexing system remains as a research goal for the long-term.

8 Conclusion

The CBR-HS classification system is being incrementally built, and it will continue to be refined as we add papers. The different categories proved useful for indexing and tracking CBR-HS core research literature. With its system of tiers, some of which may be omitted, this system is very flexible and can index either fielded applications, frameworks, or survey papers. This study has identified interesting and major research themes and trends characteristic of applied domains such as health sciences domains. I has also compared these themes with those in the CBR conference literature, and found both common elements and differences. This analysis of CBR-HS literature also permits to identify potential future research directions. Future directions include visualization and evolution tracking of CBR-HS literature, comparison with automatic classification, as well automatizing the indexing system as much as feasible.

References

1. Bichindaritz, I., Reed, J.: Methodology for Classifying and Indexing Case-Based Reasoning Systems in the Health Sciences. In: Florida Artificial Intelligence Research Society Conference, North America (March 2009), <http://www.aaai.org/ocs/index.php/FLAIRS/2009/paper/view/144>
2. Macura, R.T., Macura, K.J.: Case-based reasoning: opportunities and applications in health care. *Artificial Intelligence in Medicine* 9(1), 1–4 (1997)
3. Bichindaritz, I.: Case-based reasoning in the health sciences. *Artificial Intelligence in Medicine* 36(2), 121–125 (2006)
4. Bichindaritz, I., Montani, S.: Advances in case-based reasoning in the health sciences. *Artificial Intelligence in Medicine* 51(2), 75–79 (2011)
5. Bichindaritz, I., Marling, C.: Introduction to the special issue on case-based reasoning in the health sciences. *Computational Intelligence* 22(3–4), 143–147 (2006)
6. Bichindaritz, I., Montani, S.: Introduction to the special issue on case-based reasoning in the health sciences. *Computational Intelligence* 25(3), 161–164 (2009)
7. Bichindaritz, I., Montani, S., Portinale, L.: Special issue on case-based reasoning in the health sciences. *Applied Intelligence*, 1–3 (2008)

8. Greene, D., Freyne, J., Smyth, B., Cunningham, P.: An Analysis of Research Themes in the CBR Conference Literature. In: Althoff, K.-D., Bergmann, R., Minor, M., Hanft, A. (eds.) ECCBR 2008. LNCS (LNAI), vol. 5239, pp. 18–43. Springer, Heidelberg (2008)
9. Gierl, L., Bull, M., Schmidt, R.: CBR in Medicine. In: Lenz, M., Bartsch-Spörl, B., Burkhard, H.-D., Wess, S. (eds.) Case-Based Reasoning Technology. LNCS (LNAI), vol. 1400, pp. 273–297. Springer, Heidelberg (1998)
10. Schmidt, R., Montani, S., Bellazzi, R., Portinale, L., Gierl, L.: Case-based reasoning for medical knowledge-based systems. *International Journal of Medical Informatics* 64(2-3), 355–367 (2001)
11. Bichindaritz, I., Marling, C.: Case-based reasoning in the health sciences: what's next? *Artificial Intelligence in Medicine* 36(2), 127–135 (2006)
12. Nilsson, M., Sollenborn, M.: Advancements and trends in medical case-based reasoning: An overview of systems and system development. In: Proceedings of the 17th International FLAIRS Conference, Special Track on Case-Based Reasoning, pp. 178–183. AAAI (2004)
13. Holt, A., Bichindaritz, I., Schmidt, R., Perner, P.: Medical applications in case based reasoning. *The Knowledge Engineering Review* 20(3), 289–292 (2007)
14. Greene, D., Freyne, J., Smyth, B., Cunningham, P.: An Analysis of Research Themes in the CBR Conference Literature. In: Althoff, K.-D., Bergmann, R., Minor, M., Hanft, A. (eds.) ECCBR 2008. LNCS (LNAI), vol. 5239, pp. 18–43. Springer, Heidelberg (2008)
15. Begum, S., Ahmed, M.U., Funk, P., Xiong, N., Folke, M.: Case-Based Reasoning Systems in the Health Sciences: A Survey of Recent Trends and Developments. *IEEE Transactions on Systems, Man, and Cybernetics* 41(4), 421–434 (2011)
16. Lipscomb, C.E.: Medical subject headings (mesh). *Bulletin of the Medical Library Association* 88(3), 265–266 (2000)
17. Porter, B.W., Bareiss, E.R., Holte, R.C.: Concept learning and heuristic classification in weak-theory domains. *Artificial Intelligence* 45(1-2), 229–263 (1990)
18. Koton, P.: Reasoning about evidence in causal explanations. In: Proceedings of AAAI 1988. Seventh National Conference on Artificial Intelligence, pp. 21–26 (1988)
19. Bareiss, E.R., Porter, B.W., Wier, C.C.: Protos: an exemplar-based learning apprentice. In: Proceedings of the Fourth International Workshop on Machine Learning, pp. 12–23. Morgan Kaufmann, Los Altos (1987)
20. Bichindaritz, I., Kansu, E., Sullivan, K.M.: Case-Based Reasoning in CARE-PARTNER: Gathering Evidence for Evidence-Based Medical Practice. In: Smyth, B., Cunningham, P. (eds.) EWCBR 1998. LNCS (LNAI), vol. 1488, pp. 334–345. Springer, Heidelberg (1998)
21. Kolodner, J.L., Kolodner, R.M.: Using experience in clinical problem solving: introduction and framework. *IEEE Transactions on Systems, Man, and Cybernetics* 17(3), 420–431 (1987)
22. Grimnes, M., Aamodt, A.: A Two Layer Case-Based Reasoning Architecture for Medical Image Understanding. In: Smith, I., Faltings, B.V. (eds.) EWCBR 1996. LNCS, vol. 1168, pp. 164–178. Springer, Heidelberg (1996)
23. Perner, P.: An architecture for a CBR image segmentation system. *Journal on Engineering Application in Artificial Intelligence* 12, 749–759 (1999)

Research on Application of Data Mining Methods to Diagnosing Gastric Cancer

Arnīs Kiršners¹, Serge Parshutin¹, and Marcis Leja²

¹ Riga Technical University, Institute of Information Technology, 1 Kalku str., LV-1658, Riga, Latvia

² University of Latvia, Faculty of Medicine; Riga Eastern Clinical University Hospital, Digestive Diseases Centre GASTRO, 6 Linežera str., LV-1006, Riga, Latvia
{arnis.kirsners, serge.parshutin}@rtu.lv,
cei@latnet.lv

Abstract. Constantly evolving technologies bring new possibilities for supporting decision making in different areas - finance, marketing, production, social area, healthcare and others. Decision support systems are widely used in medicine in developed countries and show positive results. This research reveals several possibilities of application of data mining methods to diagnosing gastric cancer, which is the fourth leading cancer type in incidence after the breast, lung and colorectal cancers. A simple decision support system model was introduced and tested using gastric cancer inquiry form statistical data. The obtained results reveal both the benefits and potential of application of DSS aimed to support a medical expert decision, and some shortcomings mainly connected with performing an appropriate data preprocessing before mining knowledge and building the model. The paper presents the technologies behind the DSS and shows the detailed evaluation process with discussions.

Keywords: gastric cancer, decision support system, data mining.

1 Introduction

Cancer is the worldwide problem in social health and one of the leading causes of death. Nevertheless it is known that the most of a cancer types are treatable. Referencing the World Health Organization data, at least 40% of all local cancer types are treatable and can be prevented, avoiding the risk factors, common not only for cancer, but also for the most chronic diseases. These risk factors are known and the most important of them are smoking, alcohol and other pernicious habits, activity shortage, adiposis (excessive weight) and different infectious agents. New medical technologies, new medicaments, vaccines, screening systems are continuously developed and introduced, all aimed at identification and treatment of cancer at initial stages, at improvement of life quality and life length for patients with cancer.

Most of the patients recourse to the experts having symptoms of the last stages of a disease, which significantly limits the list of possible treatments, thus having a negative impact on life length of a patient. People are too timid to

discuss their problems and recourse to experts having the disease symptoms with pain, fluxes, etc. In order to reveal a possible morbidity, a set of actions should be taken, which would contribute to early diagnosis of disease.

Even though globally the gastric cancer incidence is declining and in many Western countries the disease is not considered among the major health issues any more, globally the cancer of the stomach is still continuing to be an important healthcare problem. Gastric cancer is remaining the second leading cause of mortality worldwide within the group of malignant diseases after the lung cancer, and is accounting for almost 10% of cancer related deaths. Among men gastric cancer is the second (after lung cancer), but among women - the third leading (after breast and lung) cause of cancer-related deaths [12].

Today gastric cancer is the fourth leading cancer type in incidence (after the breast, lung and colorectal cancers). Close to a million new gastric cancer cases are diagnosed annually (989600 cases as reported by International Agency for the Research on Cancer (IARC) in 2008) [6]. The overall prognosis of the disease is remaining poor. The survival is closely related to the extent of the disease. If the disease is diagnosed at advanced stage, the survival is in general low. If an early cancer is diagnosed confined to the inner lining of the stomach wall, 95% 5-year survival could be reached [3].

Gastric cancer is well diagnosed using the upper endoscopy, however this is not a cheap type of analysis, thus there is a need for a decision support system for an earlier diagnosis, which would supply an expert with additional information for choosing whether the endoscopy should be performed in a specific case. The present work is a pilot research and discusses a possibility of using data mining methods for separating patients, who do need an endoscopy to be made from those, whom endoscopy is not obligate. Section 2 presents a model of such decision support system, showing its structure and describing inner processes. The experimental results are described and analysed in Section 3, followed by conclusions.

2 Model of the Decision Support System

The main objective of the proposed decision support system is to support a medical expert with additional information, helping him/her to make a decision whether a patient needs an endoscopy. It should be noted that a sphere of possible applications of such decision support systems is not limited to only diagnosing a gastric cancer. In most of developed countries decision support systems are widely used in medicine and other areas.

The decision support system contains two main modules - Data Mining module and Decision Support module (see Figure 1). The data preprocessing block is placed outside the DSS. The data preparation is an obligate process, but not necessary as part of a decision support system - the data preprocessing can be made outside DSS with any other tool available, however this does not decline the inclusion of data preprocessing module as part of DSS. Speaking about the medical data preprocessing, it should be noted that in the most cases classes in

the dataset will be highly imbalanced, causing a high increase in a false negative rate. One of the solutions to this problem is data sampling - creating a subset(s) of data, where classes are more balanced, as compared to the initial dataset. Different sampling models exist - static, dynamic, active sampling [1], proposing different ways to choose examples. Another option for taking on the imbalanced classes is the distributed data mining, aggregating several models in order to gain a more precise result [1]. Besides sampling, the data preprocessing should include feature selection and transformation, as in most cases exclusion of less informative attributes increases an efficiency of the system [5,9].

Data mining module contains tools for mining relationships in data and building the knowledge base for further application; it receives preprocessed statistical data and builds a relationship model, which is then saved in the knowledge base. In our specific case, the classification methods were chosen among other knowledge mining techniques. Classification model may contain a single classifier or a

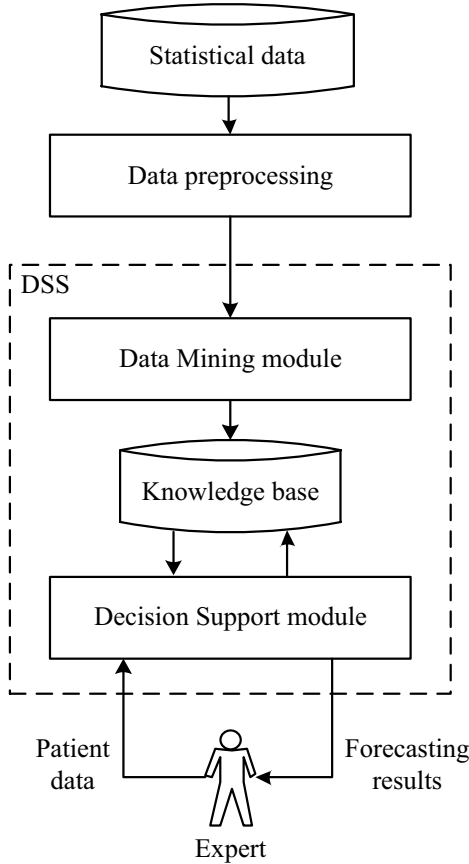


Fig. 1. Model of the decision support system

set of classifiers acting as a single one. The crisp or fuzzy classification may be applied, depending on a defined objective that should be gained.

Decision support module is the one that interacts with the user. Figure 1 shows an interaction process - an expert sends the patient data obtained from an inquiry form or via a direct contact with a patient, and receives an estimation of possible outcomes for a specific case, and then makes a final decision. The estimation of the outcomes is obtained using the knowledge base containing not only the rule sets and probability estimations (Naive Bayes), but also the efficiency coefficients for each classifier used, including classification accuracy, sensitivity (true positive rate), specificity (true negative rate) and a false-negatives rate. This coefficients can be used to show an expert the confidence of the forecast made by the decision support system.

The model of the decision system is simple and client oriented - it does not need a medical expert to have an advanced knowledge in statistics or data analysis, making it simple in application.

3 Experimental Results

In the previous section it was mentioned that in the current work the DSS is using the classification methods to mine knowledge from data. Three classification algorithms were chosen for evaluation - the Bayesian classification algorithm (Naive Bayes) [4, 5, 11], the decision tree classifier C4.5 [4, 5, 10, 11] and the classification rule induction algorithm CN2 [2, 7, 8]. All of the classifiers are known and the supplied references have a description of algorithms behind each classifier. Those three classifiers were chosen for the pilot research oriented to define whether or not simple classification algorithms can be used to process with a small dataset with class dominance. Other methods like SVMs or Nearest Neighbour classifiers were not used as the dataset contained mostly discrete attributes and was relatively small for application of SVMs. The experiments were performed using the medical data of patients who filled the gastric cancer inquiry form. The dataset contained 819 examples where 24 examples (3%) with positive diagnosis and 795 negative examples (97%) where described by 31 attribute - ID, target attribute and 29 descriptive attributes.

The diagnosis for each patient was assigned using the endoscopy, and it can be seen that in most cases the endoscopy was not necessary, as the final result was negative. The main objective of the proposed system is to lessen the false positive rate, simultaneously maintaining high sensitivity. Returning to the initial dataset the classes are highly imbalanced, which may lead to incorrectly interpreted results. All three classifiers were trained using all 819 examples with full feature set and tested using the 10-fold cross-validation; the results of experiments are given in Table 1.

All classifiers show a classification accuracy greater than 95%, but the sensitivity (true positive rate) of a target class - positive diagnosis, remains 0 or close to it, pointing out that classifiers were not able to correctly classify examples with a positive diagnosis. Such results are common for datasets with highly imbalanced classes, as classifiers perceive class significance equally weighted, thus

Table 1. Classification results with full dataset

Algorithm	CA	Sensitivity	Specificity
Naive Bayes	0.968	0.000	0.997
C4.5	0.957	0.042	0.985
CN2 Rules	0.951	0.000	0.980

in most classes a single rule is made - classify each record as one with dominant class. Different options are available for improving the classification efficiency - application of cost matrix in training stage, using negative selection (anomaly detection) instead of common classification, data synthesis, feature selection, data sampling and others. In this research the feature selection and the data sampling options were applied. First, the correlation analysis [5, 9, 11] of 29 available descriptive attributes was made and 10 attributes were selected. Table 2 summarizes the results of an attribute correlation analysis.

Table 2. Attribute correlation analysis

No.	Attribute	<i>F</i> -value	<i>p</i> -value
1	Weight loss (T/F)	3.7813	0.0521
2	Age (years)	3.1399	0.0009
3	Weight loss in last 6 months (kg)	2.7813	0.0168
4	Vomiting (T/F)	1.7794	0.1825
5	Relatives have other tumours (T/F)	1.7006	0.1825
6	Constipations (T/F)	1.5049	0.2202
7	Heartburn with proximal spreading (T/F)	1.4589	0.2274
8	First-degree relatives have gastric cancer (T/F)	1.3195	0.2510
9	Cigarettes per day	1.2667	0.2762
10	Flatulence (T/F)	1.2174	0.2701

The list of selected attributes was reviewed by our medical expert and it was stated that at least two attributes - "Weight loss" and "Weight loss in last 6 months", can be removed from the list. It has been pointed out that if a patient has an unplanned weight loss, the additional laboratory analysis (endoscopy) will always be performed. Thus the two mentioned attributes were removed from our list, leaving eight attributes for further analysis. It was decided to perform data sampling in two different proportions, shown in Table 3. Five different datasets were randomly generated using each proportion, no duplication was applied.

All generated datasets contain all positive examples, available in an initial dataset, and randomly selected negative examples, the number of which is set using the defined proportion. The number of examples is relatively small and can

Table 3. Description of generated subsets

No.	Nr. of datasets	Proportion of classes	Positive ex.	Negative ex.	Total ex.
1	5 sets	1 x 2	24	48	72
2	5 sets	1 x 4	24	96	120

decrease the confidence of results in case if the cross-validation is applied, thus each of ten generated datasets was randomly split into training and testing sets using the 70% for training and 30% for testing [5,11]. The proportions of classes in train and test sets remained the same as in the subset before splitting (see Table 3). The experimental results are presented in the next two subsections - Subsection 3.1 summarizes the evaluation results, obtained using the No.1 datasets; the results using the No.2 datasets are given in Subsection 3.2.

3.1 Experimental Results Using No.1 Datasets

Each of the three classifiers was trained and tested using each of prepared datasets. In order to confirm that feature selection can increase an efficiency of the system, some experiments were performed with three different feature sets:

- First feature set with all 29 descriptive attributes;
- Second feature set with eight attributes from Table 2 excluding attributes 1 and 3;
- Third feature with six attributes, obtained by excluding from Second feature set attributes "Cigarettes per day" and "Flatulence" (see Table 2).

Table 4 shows the result obtained using the First feature set and training and testing each classifier with all five subsets. It can be seen that the average sensitivity of classifiers increased, as compared to the data in Table 1, but still remains less than 50%. The increase in sensitivity shows that changing the proportions of classes the classifier s were forced to create relationship model, containing both classes, however the results are highly dependent on the subset and the variation in sensitivity confirms it.

Table 5 shows the classification accuracy, sensitivity and specificity of classifiers, obtained using the Second feature set. The average sensitivity increased and the false-negatives rate decreased, comparing to the results in Table 4. The average specificity of each classifier remains high, as also the average sensitivity is greater than 50%, but still highly varies from set to set.

Table 6 provides evaluation results using the Third feature set with six attributes. The average results decreased, comparing to data in Table 5, showing that attributes 9 and 10 - the number of cigarettes smoked per day and the flatulence, should not be excluded from feature set.

Figure 2 depicts the average values of classification accuracy, sensitivity and specificity for all classifiers separately for each feature set. It can be seen that all

Table 4. Evaluation results using the First feature set

CA	Set 1	Set 2	Set 3	Set 4	Set 5	Average	St.dev.
Naive Bayes	0.64	0.73	0.73	0.68	0.50	0.655	0.084
C4.5	0.64	0.73	0.59	0.55	0.64	0.627	0.060
CN2 Rules	0.59	0.82	0.59	0.59	0.77	0.673	0.101
Sensitivity							
Naive Bayes	0.43	0.71	0.29	0.43	0.27	0.429	0.156
C4.5	0.71	0.71	0.14	0.27	0.57	0.486	0.232
CN2 Rules	0.27	0.86	0.00	0.43	0.43	0.400	0.277
Specificity							
Naive Bayes	0.73	0.73	0.93	0.80	0.60	0.760	0.108
C4.5	0.60	0.73	0.80	0.67	0.67	0.639	0.068
CN2 Rules	0.73	0.80	0.87	0.67	0.93	0.800	0.094
False-Negatives rate							
Naive Bayes	0.27	0.15	0.26	0.25	0.36	0.258	0.065
C4.5	0.18	0.15	0.33	0.33	0.23	0.247	0.075
CN2 Rules	0.31	0.07	0.35	0.29	0.22	0.249	0.096

Table 5. Evaluation results using the Second feature set

CA	Set 1	Set 2	Set 3	Set 4	Set 5	Average	St.dev.
Naive Bayes	0.64	0.86	0.68	0.68	0.64	0.700	0.084
C4.5	0.73	0.68	0.55	0.55	0.64	0.627	0.073
CN2 Rules	0.77	0.86	0.59	0.59	0.77	0.718	0.109
Sensitivity							
Naive Bayes	0.57	1.00	0.43	0.43	0.57	0.600	0.210
C4.5	0.71	0.86	0.29	0.29	0.57	0.543	0.229
CN2 Rules	0.43	0.86	0.43	0.43	0.43	0.514	0.171
Specificity							
Naive Bayes	0.67	0.80	0.80	0.80	0.67	0.747	0.065
C4.5	0.73	0.60	0.67	0.67	0.67	0.667	0.042
CN2 Rules	0.93	0.87	0.67	0.67	0.93	0.813	0.122
False-Negatives rate							
Naive Bayes	0.23	0.00	0.25	0.25	0.23	0.192	0.097
C4.5	0.15	0.10	0.33	0.33	0.23	0.23	0.094
CN2 Rules	0.22	0.07	0.29	0.29	0.22	0.217	0.078

Table 6. Evaluation results using the Third feature set

CA	Set 1	Set 2	Set 3	Set 4	Set 5	Average	St.dev.
Naive Bayes	0.68	0.73	0.73	0.50	0.55	0.636	0.095
C4.5	0.73	0.73	0.82	0.68	0.73	0.736	0.045
CN2 Rules	0.68	0.82	0.73	0.68	0.77	0.736	0.053
Sensitivity							
Naive Bayes	0.71	0.86	0.029	0.00	0.57	0.486	0.308
C4.5	0.43	0.86	0.43	0.00	0.43	0.429	0.270
CN2 Rules	0.14	0.71	0.14	0.00	0.43	0.268	0.256
Specificity							
Naive Bayes	0.67	0.67	0.93	0.73	0.53	0.707	0.131
C4.5	0.87	0.67	1.00	1.00	0.87	0.880	0.122
CN2 Rules	0.93	0.87	1.00	1.00	0.93	0.947	0.050
False-Negatives rate							
Naive Bayes	0.17	0.09	0.26	0.39	0.27	0.236	0.101
C4.5	0.24	0.09	0.21	0.32	0.24	0.218	0.073
CN2 Rules	0.30	0.13	0.29	0.32	0.22	0.252	0.068

classifiers have shown an increase in average sensitivity using the Second feature set with eight attributes and the value remains greater than 50%, however the average false negative rate remains above the 20% level.

The results of experiments with No.1 datasets (see Table 3) showed that sampling and feature selection can increase an efficiency of classifier, however the results show a high variance in estimations, especially in sensitivity. The main reason of that is the small number of examples in each subset, comparing to the initial dataset. Nevertheless individual results with sensitivity higher than 70% were reached.

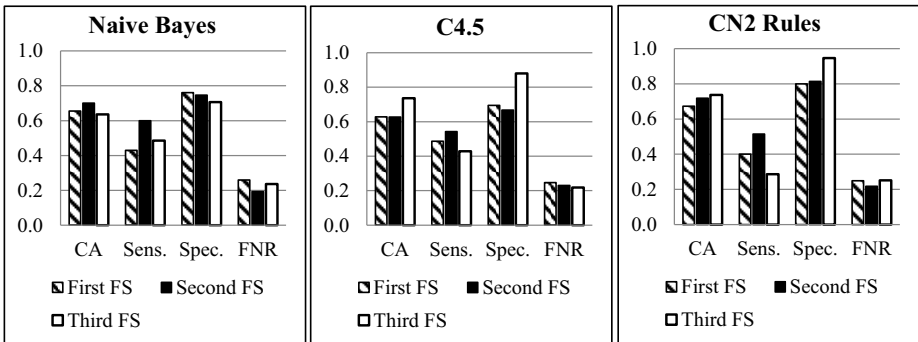


Fig. 2. Average values for classifiers in experiments with No.1 datasets

3.2 Experimental Results Using No.2 Datasets

This subsection shows the experimental results obtained using the No.2 datasets (see Table 3). The same feature sets were used as in previous subsection. Table 7 shows the evaluation results using the First feature set. As compared to the results for the same feature set, obtained using the No.1 datasets, the present results are significantly lower - the increase in negative class examples resulted in efficiency recession for all classifiers.

Table 7. Evaluation results using the First feature set

CA	Set 1	Set 2	Set 3	Set 4	Set 5	Average	St.dev.
Naive Bayes	0.75	0.81	0.78	0.72	0.69	0.750	0.039
C4.5	0.58	0.78	0.69	0.69	0.61	0.672	0.069
CN2 Rules	0.64	0.83	0.78	0.75	0.69	0.739	0.067
Sensitivity							
Naive Bayes	0.00	0.14	0.00	0.14	0.14	0.086	0.070
C4.5	0.14	0.29	0.14	0.14	0.29	0.200	0.070
CN2 Rules	0.14	0.57	0.14	0.14	0.00	0.200	0.194
Specificity							
Naive Bayes	0.93	0.96	0.96	0.86	0.83	0.910	0.056
C4.5	0.69	0.89	0.83	0.83	0.69	0.786	0.083
CN2 Rules	0.76	0.89	0.93	0.89	0.86	0.869	0.059
False-Negatives rate							
Naive Bayes	0.21	0.18	0.20	0.19	0.20	0.195	0.010
C4.5	0.23	0.16	0.20	0.20	0.20	0.198	0.022
CN2 Rules	0.21	0.10	0.18	0.19	0.22	0.181	0.041

The results obtained using the Second feature set are given in Table 8. The efficiency of classifiers is better than in the case of using the First feature set (see Table 7), but still lower comparing to experimental results with the No.1 datasets.

The results of the final set of experiments with No.2 datasets using the Third feature set with six attributes, are shown in Table 9, the efficiency recession remains, showing that sampling process is highly target specific and, if used improperly, can decrease efficiency of a classifier.

Figure 3 depicts the average values of classification accuracy, sensitivity and specificity for all classifiers separately for each feature set, training and testing classifiers on Nr.2 subsets. Comparing to the results in Figure 2, the only measure that improved is the false-negatives rate.

3.3 Evaluation of the Best Classification Model Obtained

Analysing results obtained in Subsection 3.1 and 3.2, it was decided to evaluate on the initial dataset the best classification models that were obtained using

Table 8. Evaluation results using eight attributes

CA	Set 1	Set 2	Set 3	Set 4	Set 5	Average	St.dev.
Naive Bayes	0.61	0.89	0.83	0.72	0.75	0.761	0.096
C4.5	0.61	0.86	0.69	0.81	0.78	0.750	0.088
CN2 Rules	0.69	0.64	0.81	0.81	0.81	0.750	0.070
Sensitivity							
Naive Bayes	0.29	0.43	0.14	0.14	0.29	0.257	0.107
C4.5	0.14	0.29	0.14	0.57	0.29	0.286	0.090
CN2 Rules	0.43	0.14	0.14	0.00	0.29	0.200	0.146
Specificity							
Naive Bayes	0.69	1.00	1.00	0.86	0.86	0.883	0.115
C4.5	0.72	1.00	0.83	0.86	0.90	0.862	0.090
CN2 Rules	0.76	0.76	0.97	1.00	0.93	0.883	0.104
False-Negatives rate							
Naive Bayes	0.20	0.12	0.17	0.19	0.17	0.171	0.028
C4.5	0.22	0.15	0.20	0.11	0.16	0.168	0.040
CN2 Rules	0.15	0.21	0.18	0.19	0.16	0.179	0.023

Table 9. Evaluation results using six attributes

CA	Set 1	Set 2	Set 3	Set 4	Set 5	Average	St.dev.
Naive Bayes	0.61	0.86	0.83	0.78	0.81	0.778	0.088
C4.5	0.72	0.86	0.83	0.81	0.81	0.806	0.046
CN2 Rules	0.72	0.81	0.78	0.81	0.81	0.783	0.032
Sensitivity							
Naive Bayes	0.43	0.43	0.29	0.43	0.29	0.371	0.070
C4.5	0.14	0.29	0.29	0.00	0.29	0.200	0.114
CN2 Rules	0.29	0.29	0.14	0.00	0.29	0.200	0.114
Specificity							
Naive Bayes	0.66	0.97	0.97	0.86	0.93	0.876	0.117
C4.5	0.86	1.00	0.97	1.00	0.93	0.952	0.052
CN2 Rules	0.83	0.93	0.93	1.00	0.93	0.924	0.055
False-Negatives rate							
Naive Bayes	0.17	0.13	0.15	0.14	0.16	0.149	0.017
C4.5	0.19	0.15	0.15	0.19	0.16	0.169	0.021
CN2 Rules	0.17	0.16	0.18	0.19	0.16	0.172	0.015

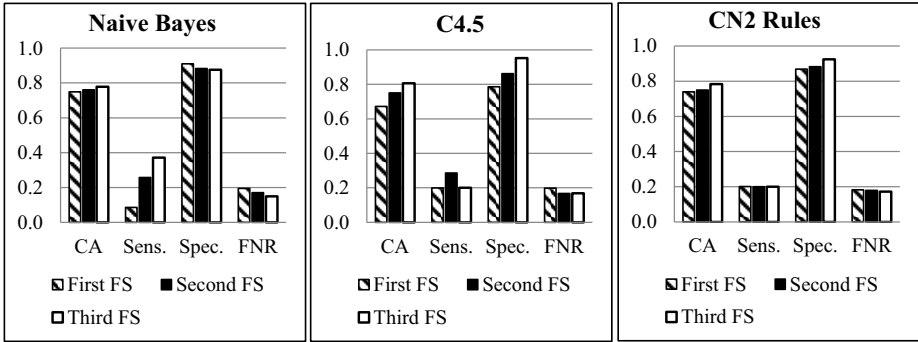


Fig. 3. Average values for No.2 datasets

sampled datasets. The higher sensitivity and the lowest false-negatives rate were reached training and testing classifiers on the second subset of the No.1 datasets, while using the Second feature set (see Table 5). The training records of the second subset were removed from the initial dataset, leaving 769 of 819 records for testing - 762 negative examples and 7 positive examples. Table 10 shows the obtained results.

Table 10. Experimental results using the best classification model

Classifier	CA	Sensitivity	Specificity	False-Negatives rate
Naive Bayes	0.86	1.00	0.80	0.00
C4.5	0.68	0.86	0.60	0.10
CN2 Rules	0.86	0.86	0.87	0.07

The obtained sensitivity of all three classifiers remained on the same level as in the results, described in Subsection 3.1 (see Table 5), however the specificity and classification accuracy decreased by 10% in general. Analysing obtained results it can be stated that classifiers show high true positive rate - sensitivity, resulting in correct diagnosis for the most of patients with gastric cancer, which is good. From the other hand, the specificity of classifiers remains on the level of 65-70%, which means that for about 30% of patients with negative diagnosis the decision support system suggested to make an endoscopy. This is a good result, comparing to the initial case, when an endoscopy was performed for each patient. Looking at the results from the other side - the false-negatives rate still remains above zero level. This means that some patients with positive diagnosis will remain unthreatened, meaning that the decision support system should not be used as a primary source for decision making. Figure 4 shows the classification tree built by the C4.5 algorithm. The tree returns good classification results, however contains some conflicting rules, like *IF Age ≤ 66 AND Relatives does*

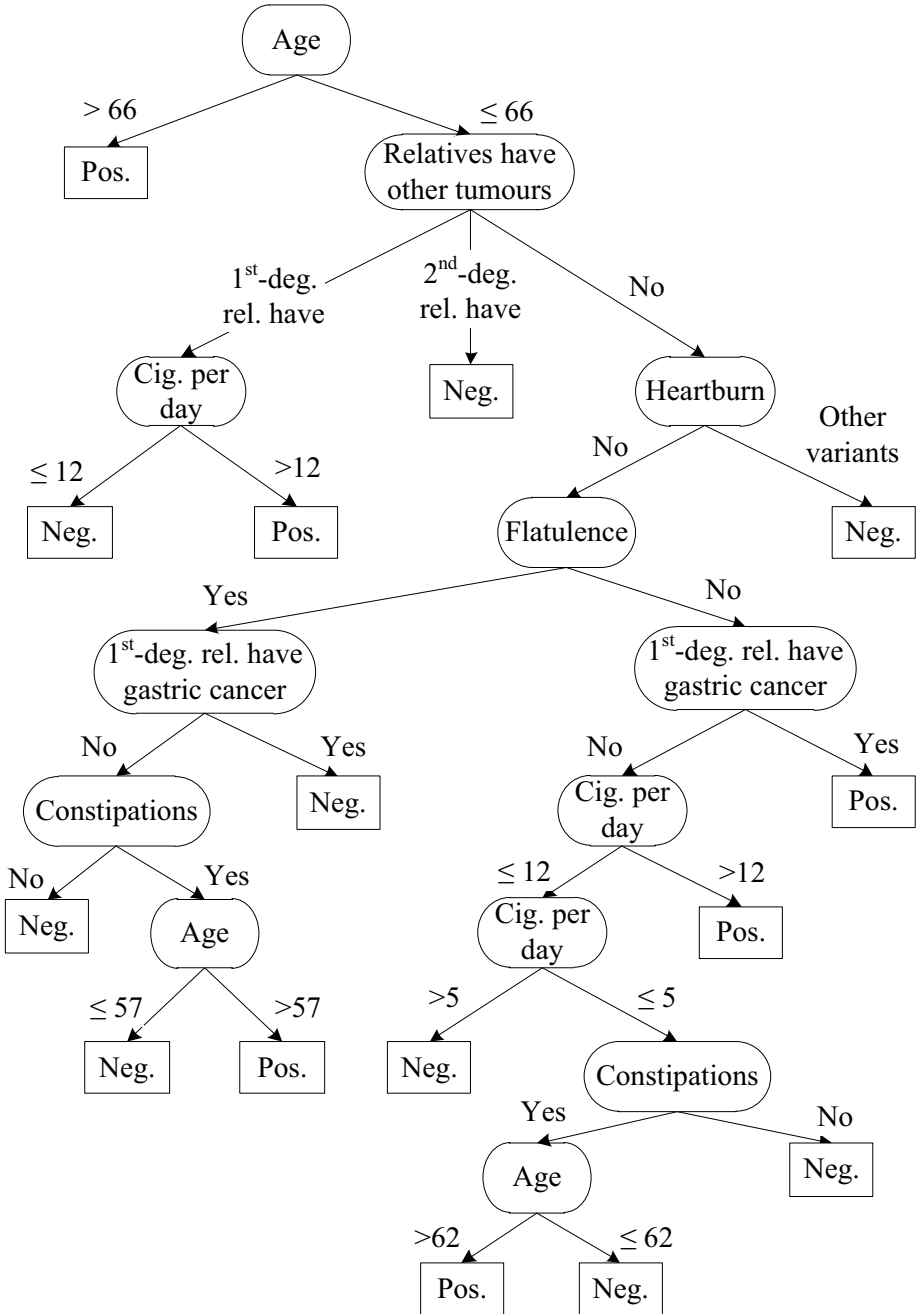


Fig. 4. Classification tree

not have other tumours AND Have no heartburn AND 1st-degree relatives have gastric cancer THEN result is Negative. This points out that classifiers are data specific and may contain rules that are normal for a training set, but do not concur with an opinion of a medical expert. That is one of the reasons for system to be a decision support not a decision system.

4 Conclusions

Gastric cancer is not the only disease the presented system can be used for. As it was mentioned earlier, the decision support systems are widely used in the healthcare. The present research showed one of possible applications of data mining methods in diagnosing the gastric cancer. The obtained results gave answers to different questions, connected with data preprocessing and especially feature selection and sampling, and defined directions for future research. The proposed decision support system is able to mine knowledge in medical data and apply it to evaluation of alternatives for each specific case. The experimental results have shown the average sensitivity greater than 50% and 86-100% at most, at the same time having classification accuracy and specificity close to 65-70% and false-negatives rate on the level of 20% on average. In comparison with an initial state when an endoscopy was performed for each patient, the application of the proposed DSS would lessen it by 70%, leaving 30% as false positives. The research in application of DSS in healthcare will be continued and for the future tasks it is planned to enlarge the initial dataset and recheck the results experimentally obtained and presented in the paper. Other option that will be considered is the application of association analysis and other anomaly detection techniques to mine knowledge in medical data.

Acknowledgements. This work has been partly supported by the European Social Fund within the project "Support for the implementation of doctoral studies at Riga Technical University".

This study was supported in part from the project of European Social Fund No. 009/0220/1DP/1.1.1.2.0/09/APIA/VIAA/016 "Multidisciplinary research group for early cancer detection and cancer prevention".

References

1. Aounallah, M., Quirion, S., Mineau, G.W.: Distributed Data Mining vs. Sampling Techniques: A Comparison. In: Tawfik, A.Y., Goodwin, S.D. (eds.) Canadian AI 2004. LNCS (LNAI), vol. 3060, pp. 454–460. Springer, Heidelberg (2004)
2. Clark, P., Niblett, T.: The CN2 induction algorithm. *Machine Learning* 3, 261–283 (1989)
3. Crew, A., Neugut, K.: Epidemiology of gastric cancer. *World Journal of Gastroenterol* 12, 354–362 (2006)
4. Dunham, M.: *Data Mining Introductory and Advanced Topics*. Prentice Hall (2003)

5. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann (2006)
6. Jemal, A., Bray, F., Center, M.M., Ferlay, J., Ward, E., Forman, D.: Global cancer statistics. *CA: A Cancer Journal for Clinicians* 61, 69–91 (2011)
7. Lavrac, N., Flach, P., Kasek, B., Todorovski, L.: Rule induction for subgroup discovery with CN2-SD. In: 2nd International Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning, pp. 77–81. University of Helsinki (2002)
8. Lavrac, N., Flach, P., Kasek, B., Todorovski, L.: Subgroup discovery with CN2-SD. *Journal of Machine Learning Research* 5, 153–188 (2004)
9. Pyle, D.: Data Preparation for Data Mining. Morgan Kaufmann (1999)
10. Ruggieri, S.: Efficient C4.5. *IEEE Transactions on Knowledge and Data Engineering* 14, 438–444 (2002)
11. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson Education (2006)
12. World Health Organization (2011), <http://www.who.int/en/>

SOHAC: Efficient Storage of Tick Data That Supports Search and Analysis

Gabor I. Nagy and Krisztian Buza

Budapest University of Technology and Economics
Magyar tudósok körútja 2, H-1117 Budapest, Hungary
gnagy@tmit.bme.hu, buza@cs.bme.hu
<http://www.bme.hu>

Abstract. Storage of tick data is a challenging problem because two criteria have to be fulfilled simultaneously: the storage structure should allow fast execution of queries and the data should not occupy too much space on the hard disk or in the main memory. In this paper, we present a clustering-based solution, and we introduce a new clustering algorithm that is designed to support the storage of tick data. We evaluate our algorithm both on publicly available real-world datasets, as well as real-world tick data from the financial domain provided by one of the world-wide most renowned investment bank. In our experiments we compare our approach, SOHAC, against a large collection of conventional hierarchical clustering algorithms from the literature. The experiments show that our algorithm substantially outperforms – both in terms of statistical significance and practical relevance – the examined clustering algorithms for the tick data storage problem.

1 Introduction

In order to describe objects or phenomena in real-world applications, usually, a relatively large set of attributes (or features) are necessary. The values of these attributes often change over time, e.g., prices on the stock market, temperature and humidity of the air, blood pressure or pulse of a person, etc. In most cases, the dynamics of these attributes, i.e., how they change their values, are almost as important as (or sometimes even more important than) the current values of the attributes. Therefore, we need to keep track of the changes of those values which results in very large collections of data.

In particular, the size of data describing financial transactions of stock markets may be several tera or even petabytes. Such data is often called tick data, see e.g. [14]. Tick data can be considered as a matrix that represents trades of financial assets. Columns of this matrix correspond different properties of a transactions, such as price, volume of the trade, the symbol of an asset, etc. Every time a transaction is executed or a quote is given for a stock, a row is appended to this matrix. Therefore, this matrix grows rapidly while a technology is necessary that allows efficient storage and quick retrieval of the data. Solutions are often built over database technology such as a KDB database [1]. However, as

¹ <http://kx.com/>

we will describe in more detail, a straightforward application of such technology leads to suboptimal storage of stock market data.

The major reason for the aforementioned storage to be suboptimal is the redundancy of conventional techniques: in case of a straightforward solution, one would use orders of magnitudes more storage space (either disk space or main memory) than required, therefore, storage, access, search and analysis of the data becomes computationally more expensive than necessary. One way to alleviate this problem is the usage of regular data compression methods (see e.g. [16] for an excellent overview). This approach is well-suited for cheap storage of large, historical archives of the data. However, it does not support quick access to the data: if the data is compressed, in order to be able to execute an analytic or search query, a large archive (or at least some parts of that) must be decompressed which might be computationally expensive and therefore the procedure could become highly inefficient. This problem becomes crucial if *many* queries has to be executed which is usually the case in real-life applications, e.g. when trading on a stock market.

In this paper, we aim at finding a compromise between the both aforementioned cases, i.e., we aim at developing a storage structure for tick data that reduces the storage space required by the straightforward approach while it allows to execute search and analytic queries efficiently. In particular, our approach is based on the decomposition of a large tick data matrix into two (or three) much smaller matrices. We achieve this decomposition by the clustering of the columns of the matrix. Although, conventional clustering algorithms achieve significant improvements, motivated by hierarchical clustering algorithms, we develop a new clustering algorithms that minimize storage space required for a tick data matrix. Therefore we call our approach SOHAC, Storage-Optimizing Hierarchical Agglomerative Clustering. We evaluate SOHAC both on publicly available real-world datasets, as well as real-world tick data from the financial domain provided by Morgan Stanley, one of the world-wide most renowned investment bank. In our experiments we compare our approach against a large collection of conventional hierarchical clustering algorithms from the literature. The experiments show that our algorithm significantly – both in terms of statistical significance and practical relevance – outperforms the examined clustering algorithms for the tick data storage problem.

This paper is organized as follows: Section 2 reviews related works. Our approach is described in Section 3, followed by our experiments in Section 4. Finally, we conclude in Section 5.

2 Related Work

The availability of high-resolution data describing transactions on financial markets, especially *tick data* (also known as *tick-by-tick data*) allows thorough analysis of the markets and their dynamics. Some of the most relevant recent works focused for example on currency exchange rates [24], [15], [18], [17], stock market tick data [14], risk analysis [7] and and the dynamics of stock markets [3]. Based

on tick data, Akram et al. empirically studied the law of one price on different financial markets [2], while Ahamad et al. focused on the summarization of tick data time series [1].

Although, storage of tick data is a core component of the systems performing the above analysis tasks, none of the above works focused on how to develop storage structures for tick data. As we will demonstrate it in Section 3.1, the storage of tick data is a non-trivial task: widely-used techniques result in redundant and therefore suboptimal solutions. Conventional compression techniques, such as run-length encoding, may result in massive reduction of the required storage space. We refer to [16] for an excellent overview of conventional compression techniques. As mentioned in Section 1, such techniques are well-suited for cheap storage of large, historical archives of the data. However, they do not support quick access to the data: if the data is compressed, in order to access the data and execute an analytic or search query, a large archive (or at least some parts of that) must be decompressed which might be computationally expensive and therefore the procedure could become highly inefficient. Therefore, in contrast to the previously discussed techniques, we build our approach on clustering which is known to have a high potential to reduce both the volume of data and its redundancy.

In the last decades, very large number of clustering algorithms were developed for various tasks (see e.g. [5], [9], [11], [12], [13] and [21]). We refer to [19] and [23] for excellent surveys of clustering algorithms. Although one can achieve substantial improvements if one uses general-purpose clustering algorithms in our approach, such conventional clustering algorithms were originally not designed for storage optimization of tick data. In contrast, the clustering algorithms we propose in Section 3 directly minimize the storage space required for tick data.

In the context of data compression, Han and Yand [10] used clustering as preprocessing for conventional data compression techniques. As they perform the actual compression by a conventional compressor, 7-zip², this approach does not support fast enough execution of search and analytic queries. Instead of using clustering as a preprocessing step for standard compressors, we build our approach on the cluster-based decomposition of tick data matrices.

3 Decomposition of Tick Data Matrix Based on Clustering



In this section, we describe our approach in more detail. First, we motivate our approach with an illustrative example, then we develop a new clustering algorithm that supports efficient storage of tick data.

3.1 An Illustrative Example

Suppose that a weather station monitors features of weather conditions. In this example, such features are the temperature, humidity and pressure of the air, the

² <http://www.7-zip.org>

a)

Time	Temp. (°C)	Hum. (%)	Press. (Pa)	Wind (v) (km/h)	Wind (dir.)	Radiation	Outlook
10:21	15	20	100 200	5	SW	low	
10:22	16	20	100 200	5	SW	low	
10:38	16	30	100 100	5	SW	low	
10:40	17	30	100 100	5	SW	medium	
10:43	18	30	100 100	10	SW	medium	
10:44	18	30	100 100	15	W	medium	
10:51	18	20	100 200	15	W	medium	

b)

Time	Hum. (%)	Press. (Pa)
10:21	20	100 200
10:38	30	100 100
10:51	20	100 200






Time	Temp. (°C)	Wind (v) (km/h)	Wind (dir.)	Radiation	Outlook
10:21	15	5	SW	low	
10:22	16	5	SW	low	
10:40	17	5	SW	medium	
10:43	18	10	SW	medium	
10:44	18	15	W	medium	

Fig. 1. An illustrative example for tick data. Features describing the weather are monitored continuously. Whenever the value of one of the features changes, a new row is inserted into the recordings (see the table in the top). Decomposition of such tables by features (columns) that change their values simultaneously may substantially reduce the required storage space (see the tables in the bottom of the figure).

velocity and direction of the wind, the intensity of the radiation of the sun and the overall outlook (such as sunny, cloudy, raining or snowing). These features are monitored continuously over the time. Whenever the value of one of these features changes, new row is inserted into the recordings. This new row contains the values of the features as well as time-stamp indicating *when* the observations were made. See the matrix in the top of Figure 1.

This representation, called tick data, is well-suited for queries: for example, if we are interested for the features of the weather at 10:30 o'clock, we only need to find the row corresponding the most recent observation *before* 10:30, i.e., we have to consider the row at 10:22. This row describes the "state of the world", i.e., it contains the values of all the features that are relevant in the current application. Such queries regarding the "state of the world" at a given time can be effectively supported by indexing techniques.

The only disadvantage of the representation shown in the top of Figure 1 is that the total size of the matrix may become much larger than actually required. In order to illustrate this we stored the same information in two smaller matrices in the bottom of Figure 1. In our approach such decompositions are based on the clustering of columns: in the example, we consider two clusters of columns. One of the clusters contains *Humidity* and *Pressure*, while the other cluster contains

the other columns, i.e., *Temperature*, *Velocity of the wind*, *Direction of the wind*, *Radiation* and *Outlook*. As shown in the example, due to the decomposition, we can save storage space: the total number of cells required to store the data was reduced from $7 \times 7 = 49$ to $3 \times 2 + 5 \times 5 = 31$ (without counting the cells in the column *Time* which acts like an index column). This corresponds to a *compression ratio* of $31/49 \approx 0,633$.

While the decomposition reduces the required storage space, in the worst case, the computational complexity of a query may increase moderately: if we are interested for *all* the features describing the weather at 10:30, we have to execute two queries instead of one, however, both queries are executed on much *smaller datasets* (and therefore the overall execution time is expected to grow only moderately compared to the previous case). Whereas if we are only interested for the *temperature and radiation* we have to execute just one query on a dataset of reduced size (and therefore the overall execution time is expected to be reduced too).

The example in Figure 3.1 illustrates the decomposition of a tick data matrix in an intuitive way. Next, we systematically study such decompositions and develop an algorithm that aim at minimizing the storage space required after the decomposition.

3.2 Definitions and Problem Formulation

In general, a *tick data matrix* M is a matrix where columns correspond attributes or features while rows correspond observations of the same features at different moments of time. Rows of the matrix are ordered according to the order of observations, i.e., the values of the i -th row observed *before* the values of the j -th row if and only if $i < j$. While the observations are made, a new row is added whenever the value of an attribute changes. However, as long as none of the attribute-values change no new row is added to the matrix, therefore two rows of a tick data matrix differ in the value of at least one attribute. There is an additional column that is used to index the rows of a tick data matrix. This additional *index column* may contain, for example, ascending integer numbers (like the number of the corresponding row) or a time-stamp (see the *Time* column in the example in Section 3.1). We use the term *regular column* for all the columns other than the index column.

With *decomposition* of a tick data matrix M we mean the partitioning of the regular columns of M into k disjoint partitions P_i , $1 \leq i \leq k$, i.e., for each regular column c_j of M :

$$c_j \in P_1 \vee c_j \in P_2 \vee \dots \vee c_j \in P_k$$

and for all i, j with $i \neq j$

$$P_i \cap P_j = \emptyset.$$

Note that this partitioning refers to the regular columns only, i.e., in this formulation, the index column does not belong to any partition. Then, for each partition P_i , a matrix M_i is derived from M by selecting the index column and

those columns of M that belong to partition P_i . Subsequent rows of a derived matrix M_i may contain the same values in all the regular columns. In such cases we only keep the first row. For example, in Figure 1, $P_1 = \{\text{Humidity, Pressure}\}$, $P_2 = \{\text{Temperature, Wind (velocity), Wind (direction), Radiation, Outlook}\}$ and the corresponding matrices M_1 and M_2 are shown in the bottom left and bottom right of the Figure.

We can easily see that the original matrix can be reconstructed from the decomposition described above, and therefore, instead of the original matrix M , one can use this decomposition to calculate the results of search and analytic queries.

In this paper, we target the problem of finding a decomposition so that the required storage space is minimized. In particular, for a given number of partitions k , we aim at finding a decomposition so that the total number of all the cells in all the matrices M_i (without counting the cells in the index column) is minimized. Our approach can simply be adapted for the case of more advanced storage models, where we do not assume uniform storage cost for each cells and/or the storage costs of the index cell is also taken into account. We plan to access this issue in our future work.

We note that k is usually relatively small: for example, for the storage of tick data of financial transactions, the user is most interested for the decomposition into $k = 2$ or $k = 3$ partitions.

3.3 Clustering of Columns of Tick Data Matrix

In the literature, there are many clustering algorithms that are able to produce non-overlapping partitions in a way that these partitions together cover all the instances. Therefore, one solution for the problem defined in the previous section is to cluster the columns of a tick data matrix using one of the conventional clustering algorithms.

In the context of our problem, two regular columns are considered to be similar, if they often change values in the same row. In order to be able to reuse proximity measures from the literature, we define a *binary change indicator matrix* I over a tick data matrix M . Except the entries of the index column, all the entries of the binary change indicator matrix I are either 0 or 1 depending on whether or not the value of a cell in the tick data matrix M is equal to the value of the cell in the same column and the *previous* row of M :

$$I(i, j) = \begin{cases} M(i, j) & \text{if the } j\text{-th column is the index column in } M \\ 0 & \text{if } i > 1 \text{ and } M(i, j) = M(i - 1, j) \\ 1 & \text{otherwise} \end{cases}$$

where $M(i, j)$ and $I(i, j)$ denote the entries in the i -th row and j -th column of the tick data matrix M and binary change indicator matrix I respectively.

As an example, Figure 2 shows how the binary change indicator matrix is derived from a tick data matrix. The index column is the *Time* column in this example.

Time	Temp. (°C)	Hum. (%)	Press. (Pa)	Wind (v) (km/h)	Wind (dir.)	Radiation	Outlook
10:21	15	20	100 200	5	SW	low	
10:22	16	20	100 200	5	SW	low	
10:38	16	30	100 100	5	SW	low	
10:40	17	30	100 100	5	SW	medium	
10:43	18	30	100 100	10	SW	medium	
10:44	18	30	100 100	15	W	medium	
10:51	18	20	100 200	15	W	medium	

Time	Temp. (°C)	Hum. (%)	Press. (Pa)	Wind (v) (km/h)	Wind (dir.)	Radiation	Outlook
10:21	1	1	1	1	1	1	1
10:22	1	0	0	0	0	0	0
10:38	0	1	1	0	0	0	0
10:40	1	0	0	0	0	1	1
10:43	1	0	0	1	0	0	0
10:44	0	0	0	1	1	0	0
10:51	0	1	1	0	0	0	0

Fig. 2. Construction of a binary change indicator matrix from a tick data matrix. The tick data matrix is shown in the top of the figure, while the corresponding indicator matrix is shown in the bottom. The index column is the *Time* column in this example.

After constructing the binary change indicator matrix I , we can use its regular columns (i.e., all the columns except the index column) as instances in conventional clustering algorithms. Despite the fact that conventional clustering algorithms are not designed to produce optimal partitions in terms of our problem from Section 3.2, as we will show in the experiments, if we use the partitioning of the columns produced by conventional clustering algorithms we can achieve substantial improvements w.r.t. the required storage space compared to the case of storing the original tick data matrix. In the next section, we develop a clustering algorithm that directly optimize the storage space required to store the decomposed tick data matrix.

3.4 SOHAC: Storage-Optimizing Hierarchical Agglomerative Clustering

In this section we propose our new clustering algorithm, SOHAC, Storage-Optimizing Hierarchical Agglomerative Clustering that is designed for clustering columns of a tick data matrix. The algorithm builds on the hierarchical agglomerative strategy. Therefore, initially, all the objects belong to separate clusters. Then, clusters are iteratively merged together as long as the current number of clusters is more than k , the user-defined number of partitions. Therefore, at the end of this iterative process, k clusters are produced.

Algorithm 1. SOHAC: Storage-Optimizing Hierarchical Agglomerative Clustering for Tick Data

Require: Tick data matrix M , number of partitions k

Ensure: Partitioning of the columns of M

```

1: Construct the binary indicator matrix  $I$  from  $M$ 
2:  $P = \{\{c_1\}, \{c_2\}, \dots, \{c_n\}\}$  (Initially, each column  $c_j$  of  $M$  is a separate cluster)
3: while  $|P| > k$  do
4:    $s \leftarrow \infty$  (Storage size for the best partitioning found so far)
5:   for all pairs of clusters  $(C_i, C_j)$ , with  $C_i \in P, C_j \in P$  do
6:      $C'_i \leftarrow C_i \cup C_j$  (Merge clusters  $C_i$  and  $C_j$  into the new cluster  $C'_i$ )
7:      $P' \leftarrow P \setminus \{C_i\} \setminus \{C_j\} \cup \{C'_i\}$ 
8:      $s' =$  storage size required to store the decomposition corresponding to  $P'$ 
9:     (This can simply be computed based on  $I$ .)
10:    if  $s' < s$  then
11:       $P^* \leftarrow P'$ 
12:       $s \leftarrow s'$ 
13:    end if
14:     $P \leftarrow P^*$ 
15:  end for
16: end while
17: return  $P$ 

```

The key feature of our algorithm is that in each iteration it merges those two clusters that lead to minimal storage size of the decomposed matrix. This storage size can simply be calculated based on the binary change indicator matrix. For each examined partitioning of the columns, we decompose the binary change indicator matrix. Then, we consider the rows that contain only zeros in the regular columns. The cells of such rows can be eliminated in the examined decomposition without loss of information. Therefore, in order to determine the number of cells required for the storage of the examined decomposition, we only need to count the cells in the rows that contain only zeros in their regular columns. The pseudocode of our algorithm is shown in Algorithm 1.

4 Experiments

In this section, we describe the experiments we performed in order to evaluate our approach used and discuss the results.

4.1 Experimental Settings

Datasets — We tested our approach both on a real-world tick data describing financial transactions and several publicly available real-world dataset.

The real-world tick data from the financial domain was provided us by Morgan Stanley, one of the most renowned investment bank of the world. Therefore, in this paper, we call this dataset *MorganStanleyTickData*. MorganStanleyTickData contains 30 regular columns and 4.080.431 rows.

Additionally, we used publicly-available real-world datasets: we used some of the most popular datasets from the UCI machine learning repository [8]. In particular, these datasets were: Adult, Breast Cancer Wisconsin (Diagnostic), Car Evaluation, Forest Fires [6] and Poker Hand. As the datasets in the UCI repository do not contain tick data, in order to be able to perform reasonable experiments, as preprocessing, we removed the id values from the UCI datasets (if present) and sorted the records of the UCI datasets in lexicographical order. After sorting, the values of cells in the same columns and subsequent rows were often equal, this is the key property of tick data that our approach exploits.

Experimental Protocol — In our experiments we compared the decomposition of a tick data matrix resulting from the clusters produced by our approach, SOHAC, with the decomposition of the same tick data matrix using other clustering algorithms from the literature. We measured the quality of a decomposition by the compression ratio (CR), i.e., the ratio of the number of cells in regular columns after the decomposition and the number of cells in regular columns in the original matrix:

$$CR = \frac{\text{number of cells in regular columns after decomposition}}{\text{number of cells in regular columns in the original matrix}}$$

An example for the calculation of compression ratio can be found in Section 3.1.

We used a procedure that is similar to 10-fold-crossvalidation. In particular, we split the entire tick data matrix into 10 disjoint sub-matrices, and we repeated all the experiments 10 times: in each of the 10 rounds of the process, we used a different sub-matrix, and clustered the columns of that sub-matrix. Therefore, we could calculate the average and standard deviation of the compression ratio.

Baselines — As our approach, SOHAC, is built on hierarchical agglomerative clustering, in our experiments we focused on comparing the partitioning produced by SOHAC against the partitioning produced by different variants of hierarchical agglomerative clustering algorithms. Additionally, we compared the partitioning produced by SOHAC against the partitioning produced by k -Means [22].

Regarding the variants of hierarchical agglomerative clustering algorithms, we used Single Linkage, Complete Linkage and Average Linkage with the following proximity measures: Euclidean Distance, Cosine Similarity, Dice Similarity, Jaccard Similarity, Kulczynski Similarity, Nominal Similarity, Rogers-Taminoto Similarity, Russell-Rao Similarity Simple Matching Similarity, Chebychev Distance, Manhattan Distance and Overlap Similarity. Implementations of these proximity measures are available in the RapidMiner software too [3]. In our experiments, we used this software to calculate the partitioning with the baseline algorithms, more details about these baseline algorithms can be found in the documentation of RapidMiner and the reference therein.

³ <http://www.rapidminer.com/>

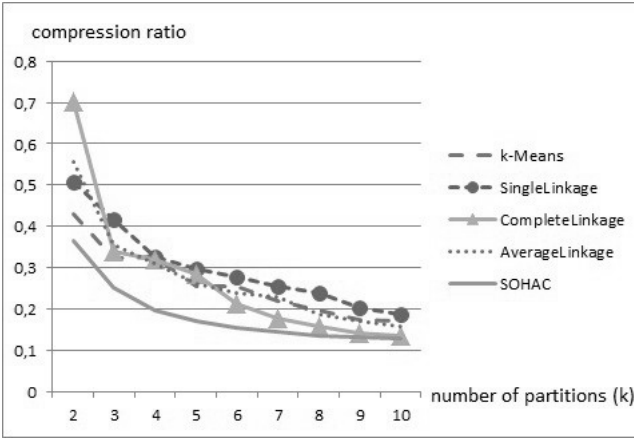


Fig. 3. Performance of our approach, SOHAC, and Complete Linkage with Euclidean distance, Single Linkage and Average Linkage with Cosine Similarity for the case of varying the number of partitions, k , between 2 and 10. The number of partitions are shown on the horizontal axis. The performance is measured in compression ratio (vertical axis) and is averaged for 10 splits.

In total, taking all the examined variants of the baselines into account, we compared our approach against 38 clustering algorithms from the literature.

4.2 Results

In our first experiment, we tested our approach on MorganStanleyTickData. We tried three different values for the number of partitions, k . Table 1 shows the results for k equal to 2, 3 and 4 respectively. Figure 3 shows the results of our approach and some of the baseline algorithms, namely k-Means and Complete Linkage with Euclidean distance, Single Linkage and Average Linkage with Cosine Similarity for the case of varying k between 2 and 10.

As we can see from Table 1, our algorithm substantially outperformed its 38 competitors. In many cases, the difference was significant in terms of average and standard deviation.

We performed our second experiment on datasets from the UCI machine learning repository. For simplicity, in Table 2, we only show the results for our approach and three baselines, Single Linkage, Complete Linkage and Average Linkage with Euclidean distance. We considered these algorithms as representatives of all the examined 38 baselines. The other examined algorithms performed similar to the ones shown in Table 2. As one can see, our approach outperformed the baselines again.

Just like in the first experiment, we additionally tested other values for the number of partitions, k , in the second experiment too. The results were similar to the ones reported in Table 2, i.e., our approach outperformed the baselines for other k values too.

Table 1. Performance of our approach, SOHAC, and the baselines on *MorganStanleyTickData*. The performance is measured by compression ratio, *smaller* values indicate *better* performance. Values are averaged for 10 splits, standard deviation is shown after the \pm symbol. For each value of k , the number of partitions, bold font denotes the winner.

Algorithm	Distance Measure	$k = 2$	$k = 3$	$k = 4$
Average-Linkage	Dice	0.9799±0.0016	0.5805±0.0596	0.3094±0.1311
	Jaccard	0.9799±0.0016	0.5805±0.0596	0.3094±0.1311
	Kulczynski	0.9799±0.0016	0.5805±0.0596	0.3094±0.1311
	Nominal	0.7385±0.1072	0.6309±0.0731	0.5612±0.0753
	Rogers-Tanimoto	0.7320±0.1032	0.6339±0.0696	0.5312±0.1184
	RussellRao	0.9799±0.0016	0.5805±0.0596	0.3094±0.1311
	SimpleMatching	0.7320±0.1032	0.6339±0.0696	0.5312±0.1184
	Chebychev	0.9799±0.0016	0.7169±0.0412	0.6836±0.0413
	Cosine	0.5556±0.2659	0.3560±0.0852	0.3084±0.0601
	Euclidean	0.7320±0.1032	0.6339±0.0696	0.5312±0.1184
	Manhattan	0.7320±0.1032	0.6339±0.0696	0.5312±0.1184
	Overlap	0.9605±0.0172	0.8788±0.0129	0.7734±0.0222
Complete-Linkage	Dice	0.7415±0.1020	0.5805±0.0596	0.3094±0.1311
	Jaccard	0.7415±0.1020	0.5805±0.0596	0.3094±0.1311
	Kulczynski	0.7415±0.1020	0.5805±0.0596	0.3094±0.1311
	Nominal	0.7044±0.0462	0.3460±0.1328	0.3254±0.1361
	RogersTanimoto	0.7013±0.0446	0.3388±0.1273	0.3190±0.1299
	RussellRao	0.9799±0.0016	0.5805±0.0596	0.3094±0.1311
	SimpleMatching	0.7013±0.0446	0.3388±0.1273	0.3190±0.1299
	Chebychev	0.9799±0.0016	0.7169±0.0412	0.6836±0.0413
	Cosine	0.8303±0.0762	0.7306±0.1298	0.3075±0.0875
	Euclidean	0.7013±0.0446	0.3388±0.1273	0.3190±0.1299
	Manhattan	0.7013±0.0446	0.3388±0.1273	0.3190±0.1299
	Overlap	0.8696±0.0475	0.7620±0.0408	0.6970±0.0441
Single-Linkage	Dice	0.9799±0.0016	0.7301±0.1952	0.3338±0.1629
	Jaccard	0.9799±0.0016	0.7301±0.1952	0.3338±0.1629
	Kulczynski	0.9799±0.0016	0.7301±0.1952	0.3338±0.1629
	Nominal	0.7607±0.1379	0.7296±0.1511	0.5612±0.0753
	RogersTanimoto	0.7520±0.1329	0.7228±0.1441	0.5587±0.0714
	RussellRao	0.9799±0.0016	0.9055±0.0264	0.4820±0.3088
	SimpleMatching	0.7520±0.1329	0.7228±0.1441	0.5587±0.0714
	Chebychev	0.9799±0.0016	0.7169±0.0412	0.6836±0.0413
	Cosine	0.5072±0.2641	0.4150±0.1893	0.3254±0.0683
	Euclidean	0.7520±0.1329	0.7228±0.1441	0.5587±0.0714
	Manhattan	0.7520±0.1329	0.7228±0.1441	0.5587±0.0714
	Overlap	0.9799±0.0016	0.9466±0.0016	0.9134±0.0018
k -Means	Euclidean	0.4291±0.1821	0.3242±0.1216	0.3244±0.1309
	Manhattan	0.8084±0.1219	0.6029±0.1224	0.4437±0.1274
SOHAC		0.3649±0.0772	0.2526±0.0587	0.1960±0.0499

Table 2. Performance of our approach, SOHAC, and Single Linkage, Average Linkage and Complete Linkage (with Euclidean Distance) on datasets from the UCI repository of machine learning datasets. The performance is measured by compression ratio, *smaller* values indicate *better* performance. Values are averaged for 10 splits, standard deviation is shown after the \pm symbol. For each dataset, bold font denotes the winner.

Dataset	SOHAC	Single Linkage	Avg. Linkage	Complete Linkage
k = 2				
Adult	0.8051±0.0256	0.8672±0.0473	0.8558±0.0408	0.8558±0.0408
Breast C.W.	0.5040±0.2420	0.5708±0.2243	0.5478±0.2181	0.5142±0.2243
Car	0.5199±0.0291	0.6347±0.0806	0.6108±0.0733	0.5909±0.0660
ForestFires	0.7816±0.0208	0.7887±0.0286	0.7834±0.0288	0.7925±0.0389
Poker Hand	0.5490±0.0001	0.7582±0.0572	0.7582±0.0572	0.7871±0.0018
k = 3				
Adult	0.7101±0.0251	0.8018±0.0515	0.7884±0.0397	0.7876±0.0388
Breast C.W.	0.4451±0.2424	0.5022±0.2167	0.4915±0.2189	0.4628±0.2292
Car	0.3869±0.0190	0.4389±0.0235	0.4391±0.0238	0.4391±0.0238
ForestFires	0.7242±0.0202	0.7406±0.0212	0.7402±0.0213	0.7387±0.0178
Poker Hand	0.4477±0.0003	0.5978±0.0011	0.5978±0.0011	0.5978±0.0011
k = 4				
Adult	0.6491±0.0222	0.7402±0.0125	0.7437±0.0215	0.7501±0.0272
Breast C.W.	0.4068±0.2344	0.4414±0.2199	0.4394±0.2215	0.4289±0.2183
Car	0.3141±0.0206	0.3146±0.0198	0.3146±0.0198	0.3146±0.0198
ForestFires	0.6857±0.0191	0.7144±0.0214	0.7113±0.0226	0.7105±0.0177
Poker Hand	0.4016±0.0004	0.4272±0.0005	0.4272±0.0005	0.4272±0.0005

The reason for the good performance of our approach is that it directly optimizes compression ratio by searching for the partitioning that corresponds to minimal storage size, while other, general-purpose clustering algorithms optimize other criteria, e.g., k-Means aims at minimizing the sum of squared distances from the centroids [19].

Additionally, we note that our partners at Morgan Stanley were extraordinarily satisfied with the results of our approach.

5 Conclusion

In this paper we focused on the storage of tick data. Our approach aimed at reducing the disk/memory occupied by the data while it allowed quick access to the data.

In particular, we developed a new clustering algorithm, SOHAC, Storage-Optimizing Hierarchical Agglomerative Clustering that is designed for partitioning the columns of a tick data matrix. This partitioning allows efficient storage of the data by the decomposition of tick data matrices. In our experiments, we compared our approach, SOHAC, against a large number of other clustering algorithms both on real-world tick data provided by Morgan Stanley and on publicly available real-world datasets from the UCI repository. The results showed

that our approach, SOHAC, substantially outperforms (in term of statistical significance and practical relevance, respectively) the examined other clustering algorithms. Furthermore, our partners at Morgan Stanley were extraordinarily satisfied with the results.

Future works may include various topics. For example, in this paper, we used a simplified model to calculate the disk/memory space required to store a tick data matrix, as we assumed uniform costs for the storage of each cell of a regular column. Additionally, one could consider other algorithms (local search, genetic algorithms, gradient descent, etc.) for finding the optimal partitioning. As a by-product of our experiments, we observed that our algorithm produced very similar clusterings on different splits of the data. This could motivate to speed-up the algorithm by sampling and the study of its stability, which could be interesting in the light of recent results concerning the theory of clustering [4]. Furthermore, one could examine whether some of the columns of the tick data matrix act as hubs and explore hub-based algorithms, such as k -Hubs [21], for the tick data storage problem. Moreover, factorization techniques, see e.g. [20], might also serve as the basis for column clustering algorithms. Last but not least, we mention that our algorithm can be applied in other domains, such as storage of multivariate time series or sensor data.

Acknowledgment. Discussions with Dr. Ferenc Bodon and Zoltan Papp, Morgan Stanley Analytics, Budapest, Hungary are greatly appreciated. The work reported in the paper has been developed in the framework of the project "Talent care and cultivation in the scientific workshops of BME" project. This project is supported by the grant TÁMOP - 4.2.2.B-10/1-2010-0009.

References

1. Ahmad, S., Taskaya-Temizel, T., Ahmad, K.: Summarizing Time Series: Learning Patterns in 'Volatile' Series. In: Yang, Z.R., Yin, H., Everson, R.M. (eds.) IDEAL 2004. LNCS, vol. 3177, pp. 523–532. Springer, Heidelberg (2004)
2. Akram, Q.F., Rime, D., Sarno, L.: Does the law of one price hold in international financial markets? evidence from tick data. *Journal of Banking & Finance* 33(10), 1741–1754 (2009)
3. Bartiromo, R.: Dynamics of stock prices. *Physical Review E* 69(6), 067108 (2004)
4. Ben-David, S., Von Luxburg, U., Pál, D.: A sober look at clustering stability. *Learning Theory*, 5–19 (2006)
5. Buza, K., Buza, A., Kis, P.: A distributed genetic algorithm for graph-based clustering. *Man-Machine Interactions* 2, 323–331 (2011)
6. Cortez, P., Morais, A.: A Data Mining Approach to Predict Forest Fires using Meteorological Data. In: *New Trends in Artificial Intelligence, Proceedings of the 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence*, pp. 512–523 (2007)
7. Dionne, G., Duchesne, P., Pacurar, M.: Intraday value at risk (ivar) using tick-by-tick data with application to the toronto stock exchange. *Journal of Empirical Finance* 16(5), 777–792 (2009)

8. Frank, A., Asuncion, A.: Uci machine learning repository (2010), <http://archive.ics.uci.edu/ml>
9. Guha, S., Rastogi, R., Shim, K.: Rock: A robust clustering algorithm for categorical attributes. *Information Systems* 25(5), 345–366 (2000)
10. Han, B., Yang, Z.: Data matrix compression by using co-clustering. In: 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2011), vol. 4, pp. 2600–2604 (July 2011)
11. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(7), 881–892 (2002)
12. Kurucz, M., Benczur, A., Csalogány, K., Lukács, L.: Spectral clustering in telephone call graphs. In: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis, pp. 82–91. ACM (2007)
13. Nanopoulos, A., Gabriel, H.-H., Spiliopoulou, M.: Spectral Clustering in Social-Tagging Systems. In: Vossen, G., Long, D.D.E., Yu, J.X. (eds.) WISE 2009. LNCS, vol. 5802, pp. 87–100. Springer, Heidelberg (2009)
14. Oh, K.J., Kim, K.: Analyzing stock market tick data using piecewise nonlinear model. *Expert Systems with Applications* 22(3), 249–255 (2002)
15. Ohnishi, T., Mizuno, T., Aihara, K., Takayasu, M., Takayasu, H.: Statistical properties of the moving average price in dollar–yen exchange rates. *Physica A: Statistical Mechanics and its Applications* 344(1), 207–210 (2004)
16. Salomon, D.: Data compression: the complete reference. Springer-Verlag New York Inc. (2004)
17. Sazuka, N.: Analysis of binarized high frequency financial data. *The European Physical Journal B-Condensed Matter and Complex Systems* 50(1), 129–131 (2006)
18. Takayasu, M., Takayasu, H., Okazaki, M.P.: Transaction interval analysis of high resolution foreign exchange data. *Empirical Science of Financial Fluctuations-The Advent of Econophysics* 18, 25 (2002)
19. Tan, P., Steinbach, M., Kumar, V., et al.: Introduction to data mining. Pearson Addison Wesley, Boston (2006)
20. Thai-Nghe, N., Drumond, L., Horváth, T., Schmidt-Thieme, L.: Multi-relational factorization models for predicting student performance. In: KDD 2011 Workshop on Knowledge Discovery in Educational Data, KDDinED 2011 (2011)
21. Tomašev, N., Radovanović, M., Mladenčić, D., Ivanović, M.: The Role of Hubness in Clustering High-Dimensional Data. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part I. LNCS (LNAI), vol. 6634, pp. 183–195. Springer, Heidelberg (2011)
22. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2011)
23. Xu, R., Wunsch, D., et al.: Survey of clustering algorithms. *IEEE Transactions on Neural Networks* 16(3), 645–678 (2005)
24. Zhou, B.: High-frequency data and volatility in foreign-exchange rates. *Journal of Business & Economic Statistics* 14(1), 45–52 (1996)

Electricity Consumption Time Series Profiling: A Data Mining Application in Energy Industry

Hongyan Liu, Zhiyuan Yao, Tomas Eklund, and Barbro Back

Turku Centre for Computer Science,
Åbo Akademi University, Department of Information Technologies
Joukahaisenkatu 3-5, 20520 Turku, Finland
{Hongyan.Liu, Zhiyuan.Yao, Tomas.Eklund, Barbro.Back}@abo.fi

Abstract. The ongoing deployment of Automated Meter Reading systems (AMR) in the European electricity industry has created new challenges for electricity utilities in terms of how to fully utilise the wealth of timely measured AMR data, not only to enhance day-to-day operations, but also to facilitate demand response programs. In this study we investigate a visual data mining approach for decision-making support with respect to pricing differentiation or designing demand response tariffs. We cluster the customers in our sample according to the customers' actual consumption behaviour in 2009, and profile their electricity consumption with a focus on the comparison of two sets of seasonal and time based variables. The results suggest that such an analytical approach can visualise deviations and granular information in consumption patterns, allowing the electricity companies to gain better knowledge about the customers' electricity usage. The investigated electricity consumption time series profiling approach will add empirical understanding of the problem domain to the related research community and to the future practice of the energy industry.

Keywords: Visual Data Mining, Clustering, Business Intelligence, Electricity Consumption Profiling, Self-Organizing Maps, Deviation Detection.

1 Introduction

Within the electricity industry, the deployment of Automated Meter Reading (AMR, i.e., remotely-readable, two-way communication smart meters) has been a topical issue for some time, especially in Europe. The progress of such deployment varies across EU countries. While Italy and Sweden have completed their nation-wide smart meter installations, and Finland is due to finish its large scale rollout to both commercial and household customers by 2013, other countries such as the UK and Belgium are still in the trial or cost-benefit analysis stage. It is well-acknowledged by the electricity industry that the deployment of smart meters and smart metering will benefit the electricity distribution business in several ways. On the one hand, short term benefits will include more efficient and accurate billing, customer services, fault detection and automated healing, just to name a few, while in the long run, it could facilitate the development of

smart grids, the integration of renewable energy resources (in particular, distributed generation), and ultimately the improvement of energy efficiency. On the other hand, the sheer amount of half-hourly or hourly measured electricity consumption data also introduces both opportunities and challenges for the electricity distribution system operators (DSOs) and /or the electricity retailers, in terms of how to manage and fully utilise such a wealth of data. So far, despite that there are successful business cases from Enel in Italy and Vattenfall Networks in Finland (Cotti and Millan 2011; Garpetun 2011), the utilisation of smart meter data or smart metering is limited to either enhancing distribution operation (e.g., automated fault detection and healing) or cost-saving from manual customer meter reading. For example, Mutanen et al. (2008) presented a method for AMR data to be used to enhance distribution state estimation. Moreover, the utilisation of AMR measurements in improving the accuracy of load modelling has been studied (Mutanen et al. 2011). Several similar studies (Abdel-Aal 2004; Charytoniuk and Chen 2000; Valtonen et al. 2010) have focused on AMR-based short-term load forecasting.

Nonetheless, according to a recent report by CEER (Council of European Energy Regulators), among the three European countries who have made decision to roll out smart meters (i.e., Italy, Sweden, and Finland), none have a demand response scheme based on smart metering. According to CEER's definition, demand response is about "*Changes in electric usage by end-use customers/micro generators from their current/normal consumption/injection patterns in response to changes in the price of electricity over time, or to incentive payments designed to adjust electricity usage at times of high wholesale market prices or when system reliability is jeopardized. This change in electric usage can impact the spot market prices directly as well as over time*" (CEER 2011). This implies that the establishment of a demand response electricity retail market not only requires the electricity end users' active engagement, but also the electricity utilities' capability for incentive pricing is crucial. To this end, we believe that in order to fully utilise the business potential enabled by smart metering technologies, it requires that the DSOs or the electricity retailers have good knowledge about their customers' timely electricity consumption patterns. Therefore, it is necessary to explore the smart meter data deeper for more gold nuggets.

In this paper, we investigate a visual data mining approach in the form of Self-Organizing Maps (SOM), namely electricity consumption time series profiling. We analyse quasi-daily smart meter data for approximately 12,000 customers in a Finnish region in 2009. We compare two sets of variables in terms of seasons and time bands partition. The objective is to study (1) what insightful knowledge can be detected by such a visual data mining approach; and (2) what is the added value for the business practice in applying such an analytical method for decision-making support, with respect to pricing differentiation or designing demand response tariffs. The results indicate that this analytical approach is capable of visualising deviations and more detailed information regarding customer's consumption patterns, which could support the case company in pricing decision-making. As will be illustrated in the following paper, this study will contribute with empirical understanding of the problem domain to the related research community and to the future practice of the electricity industry.

The paper is organised as follows: in the next two sections, the data mining method used in this study will be described first, followed by a brief introduction to the business case area. Thereafter, the experiment, results, and the analysis will be presented, and in the last part of this paper, the conclusion will be drawn and limitations and future research will be addressed.

2 Methodology

The Self-Organising Map (SOM) is one type of data mining technique based upon Artificial Neural Networks (ANNs). ANNs are designed to mimic the basic learning and association patterns of the human nervous system, and consist of a number of neurons (simple processors) connected by weighted connections. ANNs learn by adjusting the weight of each connection, increasing or decreasing the importance of the input (information) being transferred, until a desired output is achieved. Essentially, they are non-linear, multivariate regression techniques, better able to handle erroneous and noisy data than parametric statistical tools (Bishop 1995).

The SOM is a widely used unsupervised neural network, particularly suitable for clustering and visualisation tasks (Han and Kamber 2000; Kohonen 1997). It is capable of projecting the relationships between high-dimensional data onto a two-dimensional display (or map), where similar input records are located close to each other (Kohonen 1997). By adopting an unsupervised learning paradigm, the SOM conducts clustering tasks in a completely data-driven way (Kohonen 1997; Kohonen et al. 1996), i.e., no target outputs are required. Because of its robustness, it requires little a priori information or assumptions concerning the input data, and is more tolerant towards difficult data, including non-normal distributions, noise, and outliers, than traditional statistical tools. In other words, the SOM combines the objectives of both data and dimensionality reduction methods, as seen either in the clustering techniques (e.g., K-means) or in the visualisation techniques (e.g., Sammon's mapping) (Sarlin and Peltonen 2011). This capacity of the SOM motivated the authors to apply it in the present study. As the SOM algorithm itself is well-known, we refer readers to Kohonen (2001) for details.

The SOM has been applied as an analytical tool in finance, medicine, customer relationship management, and engineering applications (Back et al. 2001; Deboeck and Kohonen 1998; Eklund et al. 2003; Kaski et al. 1998; Oja et al. 2002; Yao et al. 2010). In particular, the SOM has been used in the energy sector for e.g., power system stability assessment, on-line provision control, load forecasting, as well as electricity distribution regulation and benchmarking (Lendasse et al. 2002; Nababushana et al. 1998; Rehtanz 1999; Riqueline et al. 2000; Liu et al. 2011).

A SOM is typically composed of two layers: an input and an output layer. Each input field is connected to the input layer by exactly one node, which is fully connected with all the nodes in the output layer (Berry and Linoff 2004; Wiskott and Sejnowski 1998). When the number of nodes in the output layer is large, the adjacent nodes need to be grouped to conduct clustering tasks. Accordingly, Vesanto and Alhoniemi (2000) proposed a two-level approach, e.g., the SOM-Ward clustering, to

perform clustering tasks. The dataset is first projected onto a two-dimensional display using the SOM, and the resulting SOM is then clustered. Several studies have shown the effectiveness of the two-level SOM, especially the superiority of the SOM-Ward over some classical clustering algorithms (Lee et al. 2006; Samarasinghe 2007).

As mentioned previously, the SOM-Ward clustering is a two-level clustering approach that combines local ordering of the SOM and Ward's clustering algorithm to determine the clustering result. Ward's clustering is an agglomerative (bottom-up) hierarchical clustering method (Ward 1963). The SOM-Ward starts with a clustering where each node is treated as a separate cluster. The two clusters with the minimum Euclidean distance are merged in each step, until there is only one cluster left on the map. The distance follows the SOM-Ward distance measure, which takes into account not only the Ward distance but also the topological characteristics of the SOM. In other words, the distance between two non-adjacent clusters is considered infinite, which means only adjacent clusters can be merged. A low SOM-Ward distance value represents a more natural clustering for the map, whereas a high value represents a more artificial clustering. In this way, the users can flexibly choose the most appropriate number of clusters for their data mining tasks.

3 The Finnish Business Case

The business case studied in this paper is provided by one DSO in Finland – Ålands Elandelslag (ÅEA, which is a non-profit ownership cooperative). ÅEA's distribution area has distinct geographical features and customer structure. Åland is an autonomous Finnish archipelago region with nearly 300 habitable islands. It is situated between mainland Finland in the east and Sweden in the west. ÅEA is responsible for the electricity distribution to 15 municipalities in Åland. Its distribution area covers 14,097 customers, of which Jomala is the largest (2,290 customers) and Sottunga is the smallest (184 customers). Its distribution power lines totalled 3,217 km in 2009, with high voltage lines (10kV) 1,163 km and low voltage lines (0.4kV) 2,054 km. Åland's geographical features determine that its economy is heavily dominated by shipping, trade, and tourism. The majority of the housing is in the form of summer cottages, detached houses, or town houses, while multi-storeyed buildings only account for a very small portion.

According to Statistics Åland, in 2009, Åland's electricity consumption by sector is as follows: Households (45.04%), Agriculture (7.01%), Industry (11.77%), Services (21.22%), and the Public Sector (14.97%), respectively. It shows that households, services, and the public sector constitute the majority in terms of electricity consumption in Åland. This differs from the electricity consumption breakdown on mainland Finland, where industry's electricity consumption amounts to 46%, whereas housing and agriculture, and services and construction, consume 29% and 22% respectively (source: Energiategollisuus).

The data investigated is from ÅEA meter reading registers in 2009. For each meter, the electricity usage is registered with 27 hours 20 minutes time intervals, due to the

AMR and communication technology adopted (Turtle Automated Meter Reading system). The Turtle AMR uses the power line for data transmission. The data is collected by a receiver installed at a substation and held until requested by a computer at the main office, then sent via SMS. Turtle AMR also calculates the highest rate of electricity usage for each meter during each 27hrs20mins interval, i.e., the Peak Load. Therefore, the data from meter reading registers includes Meter ID, Electricity Usage, Reading Time, Peak Load, and Peak Time.

The analysis is carried out with a focus on three types of consumption time series, including (i) weekdays vs. weekends consumption comparison, (ii) consumption seasonality, and (iii) load patterns at various times of the day (i.e., different time bands).

4 The Experiment

Even though the ÅEA smart meter data is not hourly measured, it is still possible to look into customers' electricity consumption patterns in terms of day-of-the-week, seasonal, and time band effects. Based on the meter register data, a great deal of data pre-processing work, including data transformation, aggregation, and normalisation, has to be performed to create customer signatures, with one record per customer and a range of variables capturing customers' demographic and consumption related features. We excluded the customers whose records included less than one year, or whose annual consumption is 0 kWh. There are in total 11,964 customers included in this study. The variables used fall into two types based upon their purpose – one type is used to describe the customer's general consumption and demographic profile, and the other is to investigate customers' weekday-weekend, seasonal, and time-band related consumption patterns. Regarding the second type, we compared two sets of variables – the first set is adopted from ÅEA's partition as weekdays/weekends for the time of week, seasons (i.e., January-April, May-September, and October-December), and day time (7:00-23:00)/night time (23:00-7:00) for every 24hrs, which can be seen in ÅEA's electricity tariff of Time rate; the second set is proposed by the authors, as weekdays/Saturday/Sunday for the day-of-the-week, seasons (i.e., Summer: March-September, and Winter: October-February), and four time bands (i.e., 6:00-9:00, 9:00-16:00, 16:00-22:00, 22:00-6:00) for every 24hrs. In total, there are 31 variables used in this analysis. The variables are described as follows:

Average Consumption (kWh) – is the customer's average consumption per 27hrs 20mins +/- 8mins.

Average Peak Load (kW) – is the customer's average peak demand in 2009, which is based on the highest load aggregated from three consecutive 20min intervals during each 27hrs 20mins period.

Electricity Rate – is the contractual electricity tariff the customer has chosen among 5 categories: Normal rate, Economic rate, Time rate, Irrigation rate, and Temporary Working rate, which are provided by ÅEA (available at http://www.el.ax/files/tariffhafte_20110101.pdf, in Swedish). Due to the previously

mentioned customer selection criteria set in pre-processing, the data records with Irrigation rate and Temporary Working rate are not included¹.

Housing Type – is based on historical statistics, provided by ÅEA as a reference variable, including 5 categorical attributes: Summer Cottage, Detached House, Town House, Multi-storeyed Building, and Others. Again, as with Electricity Rate, the data records with Housing Type as Others are not included in the final dataset.

Seasonal and day-of-the-week Consumption (kWh) – includes Weekday Consumption¹, Weekend Consumption¹, Jan.-Apr. Consumption, May-Sep. Consumption, Oct.-Dec. Consumption, which are adopted from ÅEA's Time-of-Use tariff; and Weekday Consumption², Saturday Consumption, Sunday Consumption, Winter Consumption, and Summer Consumption, which are proposed by the authors.

Time-based Peak Load (kW) – is the customer's average peak demand at various times of the day, including: Peak Load_Day, Peak Load_Night, which are based on ÅEA's electricity tariff; and Peak Load_6-9, Peak Load_9-16, Peak Load_16-22, Peak Load_22-6, which are proposed by the authors.

Time-based Peak Frequency (%) – is the percentage of peak demand occurring at different times of the day, including: Peak Frequency_Day, Peak Frequency_Night, which are based on ÅEA's electricity tariff; and Peak Frequency_6-9, Peak Frequency_9-16, Peak Frequency_16-22, Peak Frequency_22-6, which are proposed by the authors.

In this study, Viscosity SOMine v.5.0 (<http://www.eudaptics.com/>) is used to perform the visual data mining task. SOMine uses an expanding map size and the batch training algorithm, allowing for efficient training of maps (Deboeck and Kohonen 1998). The SOM-Ward clustering method is also used to identify clusters based on actual consumption behaviour, which eliminates the need for subjective identification of clusters (Vesanto and Alhomiemi 2000). The two sets of seasonal and time-based variables are normalised according to the respective average value before map training, i.e., each entry in a field is divided by the mean of the entire field (Baragoin et al. 2001; Collica 2007). The purpose is to address the relative significance of the value of a particular variable against the overall mean of that variable. For example, customers exhibiting average consumption patterns are given normalised values of 1, while a normalised value of 2 implies that their consumption amount or peak load is two times more than the average. In addition, all the variables included in the training process were scaled to comparable ranges in order to prevent variables with large values from dominating the result. Viscosity SOMine offers two forms of scaling, linear and variance scaling. Linear scaling is simply a linear scaling based upon the range of the variable, and is suggested as default when the range of the variable is greater than eight times of its standard deviation. Otherwise, variance scaling is used. In this study, range scaling was applied to the variables of Electricity Rate and Housing Type, while variance scaling was applied to the others.

¹ Categorical variables, such as Electricity Rate and Housing Type, must be split into binary dummy variables in order to be used with the SOM, as they represent nominal data with no inherent numerical order or distance.

We experimented with different combinations of parameters, and selected the map based on following criteria: average quantization error, normalized distortion measure, the meaningfulness of clusters, the visual interpretability, the smoothness of neighbourhood of each node, and the SOM-Ward cluster indicator. The map was trained using a map size of 279 nodes, a map ratio of 100:49, and a tension of 0.5. During the training process, the priority of categorical variables such as Electricity Rate and Housing Type, as well as the seasonal and time-based variables proposed by the authors, was set to 0. These variables thus have no influence on the training process. However, their distribution in each of the segments can be visualised on the map for comparison and profiling purposes.

In order to evaluate the robustness of the training method, a supervised ten-fold cross-validation was conducted. The entire training dataset was firstly partitioned into 10 subsets, then using 9 out of the 10 subsets each time to reiterate the map training with the same set of training parameters as was described above. The map selecting criteria set above can be held over the ten-fold iteration.

5 Results and Analysis

5.1 Cluster Profiles

The SOM divided the 11,964 customers into four clusters according to their consumption similarity in 2009. The SOM results can be seen in Figures 1-3. Since the warm colour code (e.g., red) in SOM map denotes high values while a cold colour code (e.g., blue) represents low values, the characteristics of each cluster (I-IV) can be easily identified, as summarised in Table 1. A description of each cluster follows:

- *Cluster I: High consumption customers*

Customers in cluster I account for 10% of total customers investigated and stand for 28.9% of the total consumption. They have the highest consumption profile (Average Consumption 63.0 kWh and Average Peak Load 5.1 kW). The proportion of customers using the Economic rate in cluster I (19%) is much higher than that of the other three clusters, although 80% of the customers still prefer the Normal rate. The majority of customers in cluster I live in detached house (88%), while 7%, 4%, and 1% of them are in summer cottages, town houses, or multi-storeyed buildings, respectively.

- *Cluster II: Medium-high consumption customers*

17% customers are in cluster II and they stand for 30.7% of the total consumption. They have the Medium-high consumption profile (Average Consumption 39.3 kWh and Average Peak Load 3.2 kW). Even though the majority housing type is detached house (75%), the proportion of summer cottage (18%) is the second highest after cluster IV. 5% of the customers in this cluster chose Economic rate, while 94% of them went for Normal rate.

- *Cluster III: Medium-low consumption customers*

Customers in cluster III account for 25.9% of the customer base and stand for 24.1% of the total consumption. They have Medium-low consumption profile

(Average Consumption 20.3 kWh and Average Peak Load 2.0 kW). The characteristics of cluster III are very similar to those of cluster II in that most of the customers (96%) use Normal rate and 76% of the customers live in detached houses. But the proportion of town house owners (12%) is the highest comparing to the other three clusters.

• *Cluster IV: Low consumption customers*

47.1% customers belong to cluster IV, which has the lowest consumption profile (Average Consumption 7.5 kWh and Average Peak Load 0.6 kW). They stand for 16.2% of the total consumption. 99% of customers in cluster IV have the Normal tariff contracts. Summer cottage (70%) is the major housing type within cluster IV, while detached house, town house, and multi-storeyed building account for 18%, 8%, and 4%, respectively.

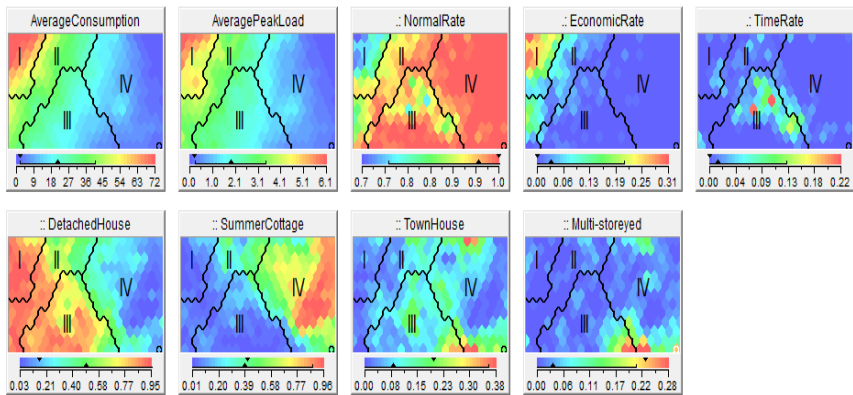


Fig. 1. Cluster profiles

5.2 Consumption Time Series Profiling

5.2.1 Weekdays vs. Weekends Consumption Comparison

Figures 2 and 3, specifically, reveal the patterns of each cluster (i.e., day-of-the-week, seasonal, and different time band consumption), and those of ÅEA's customers in general. For instance, from cluster I through cluster IV, both weekday consumption- and weekend consumption- patterns (see Figure 2) are ranging from high, medium to low, which also are in accordance with the patterns of Average Consumption in Figure 1. In addition, if comparing the consumption during weekdays/weekends (see Figure 2), or weekdays/Saturday/Sunday (see Figure 3), the patterns are nearly identical. This implies that if ÅEA intended to shift customers' demand between weekdays and weekends to mitigate system constrains or when the wholesale market price is high, ÅEA should devise enough incentive in their price signals for customers to adjust their consumption behaviour between weekdays and weekends.

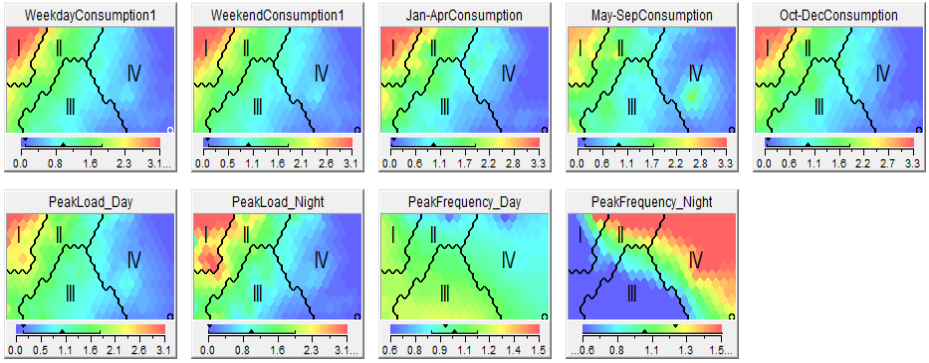


Fig. 2. Consumption patterns with AEA variables

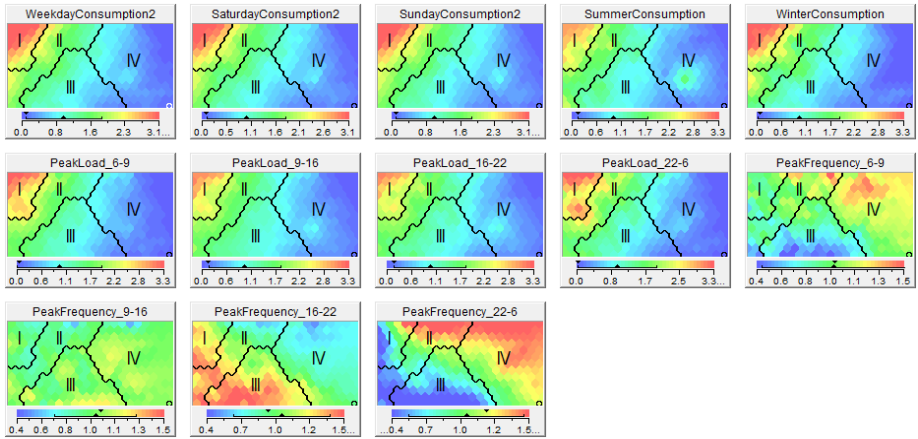


Fig. 3. Consumption patterns with proposed variables

5.2.2 Consumption Seasonality

The customers' seasonal consumption patterns vary. They follow the typical Nordic phenomena: electricity consumption is relatively higher in cold winter months than in summer time. This can be seen from both sets of seasonal consumption variables (see Figure 2 and Figure 3). However, it is important to note that there is a special group of customers in cluster IV (see Figure 4), whose electricity consumption in May-September is higher than the rest of cluster IV. This special group can be identified both from Figure 2 (May-Sep Consumption) and Figure 3 (Summer Consumption), which emphasizes that the consumption pattern deviation of this special group of customers in summer time is without regard to the summer months partition (i.e., May-Sep. as following AEA, or March-Sep. as proposed by the authors). At this point, it demonstrates that such a SOM-based data mining approach can visualize latent information for companies to take into account.

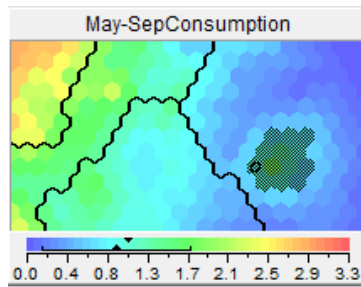


Fig. 4. Special group of customers in cluster IV

Based on the SOM visual clustering results, Figures 5, 6, and 7 summarize the comparison of various time series profiles among clusters. Figure 5 illustrates the consumption profile breakdown of each cluster and the special group within cluster IV, regarding weekday/weekend as well as seasonal consumption patterns. The different clusters have distinct consumption profiles in different seasons. For instance, regarding the Medium-low consumption customers (cluster III), their electricity usage is relatively even across different seasons (Jan-Apr., May-Sep. and Oct.-Dec.) in 2009 (red line in Figure 5). But High and Medium-high consumption customers (green and purple lines in Figure 5) had lower electricity consumption in summer time, compared to their respective cold weather seasons. On the other hand, as was pointed out before, among Low consumption customers, their May-September period consumption is relatively higher than in the rest of the seasons (see two blue lines in Figure 5).

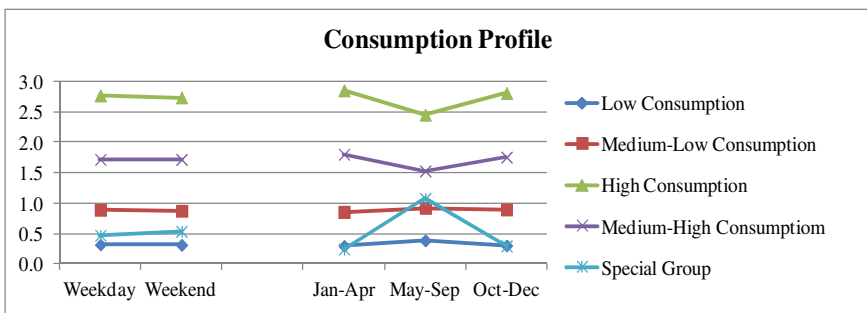


Fig. 5. Consumption profile breakdown

5.2.3 Load Patterns at Various Times of the Day

Accordingly, one can see that the patterns in terms of Peak Load at day time (7:00-23:00) and night time (23:00-7:00) (see Figure 2) are in line with the patterns of Average Peak Load in each cluster (see Figure 1). However, if examining Peak Load in four time bands in Figure 3, instead of the 2 (i.e., Day and Night) in Figure 2, slightly different picture emerges: the customers in cluster I have relatively higher peak demand in the early morning (6:00-9:00) and in the late night (22:00-6:00), compared to usual working hours (9:00-16:00) or usual peak consumption time period

(16:00-22:00). This is also represented in Figure 6, where the green line (High consumption customers of cluster I) bends up towards the ends considerably. It suggests that using the proposed four time bands can reveal more detailed information about the customers' consumption behaviour. And it might be beneficial if the company would consider using more than two time bands in their Time-of-Use pricing. The evidence can also be seen from Peak Frequency, i.e., where time-wisely speaking Peak Frequency at 6-9, 9-16, and 16-22 are equivalent to Peak Frequency_{Day}, but provide more information about consumption behaviour in different clusters. The comparison regarding how much extra information can be extracted with four time bands partition is shown in Figure 7.

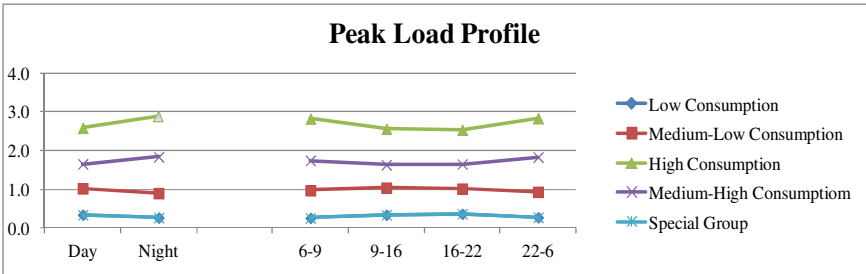


Fig. 6. Peak load profile breakdown

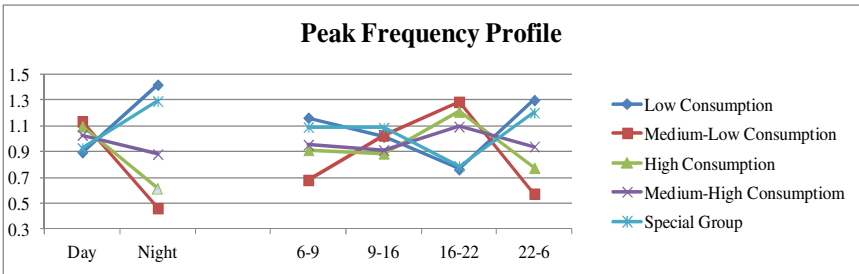


Fig. 7. Time bands partition comparison

6 Conclusion

Traditionally, the electricity utilities have classified customers according to their business nature (i.e., industrial, commercial, and residential) and their consumption bands (e.g., annual consumption < 2,000kWh, >5,000kWh, or > 18,000kWh) and housing types (e.g., detached houses, town houses, and multi-storeyed buildings) for household customers. Even in the same customer class, the consumption patterns may vary considerably due to customers' business nature / life style diversity (Keppo and Räsänen 1999). Additionally, the customer type is usually determined when the

electricity connection is contracted, which is highly likely out-dated because of later changes in the customer's profile, for example, occupancy changes in a household. Now, smart meter data provides the opportunity to group and compare the customers according to their actual energy usage, especially taking seasonal variations into account.

Enabled by the smart metering technologies and motivated by the analytical robustness of the SOM in visualisation and data exploration, in this paper we have examined a SOM-based visual data mining approach, in order to investigate how the electricity utilities can fully explore smart meter data to gain better knowledge about their customers' timely electricity consumption patterns, and in turn to support pricing decision-making. We studied a case company from Finland—ÅEA's AMR data in 2009, with the purpose of demonstrating (1) what kind of actionable knowledge the examined electricity consumption time series profiling approach can offer and (2) what is the added value for DSOs or electricity retailers in applying such a visual data mining driven analytical method in decision making support, especially with regard to pricing differentiation or dynamic pricing. First, we used the SOM to cluster 11,964 customers into four groups according to their electricity consumption similarity in 2009. Then, the consumption profile of each cluster was visualized through feature plane analysis. During the analysis we compared different variable sets in day-of-the-week, season, and time band partition, in order to extract more detailed information about the customers' consumption patterns. For instance, the result shows that there is a special customer group within the low consumption cluster IV, whose consumption pattern in summer time deviated from the rest of the cluster. Moreover, the consumption visualisation indicated that the benefit for ÅEA to design different Time-of-Use tariff on weekdays or weekends calls for a review of its pricing differentiation strategy. In addition, there is evidence that the authors' proposed four time bands could provide granular information regarding customer consumption behaviour. These findings are actionable information for the case company to take into account in their future pricing strategy making. To this end, the conclusion can be drawn that this study provides an empirical example with regard to exploring timely measured smart meter data for customer's consumption behaviour analysis. It could induce further scientific interests regarding this emerging problem domain, for example, in terms of the intersection between ubiquitous computing, data mining, and demand response simulation. It also will contribute to the future practice of the energy industry in terms of integrating data mining into their pricing decision-making support.

Nevertheless, there are limitations to this study. Firstly, it would be of great interest to compare the SOM application to using other visualisation and clustering methods such as K-means or multi-dimensional scaling methods. However, it is beyond the scope of this paper. Secondly, the scope of this analysis is determined by the specific data domain. Therefore, the discovered knowledge has its particular locality. On the other hand, such an analytical approach can be applied to other AMR data for further examination and comparison studies.

Table 1. Summary of cluster characteristics

ID	Cluster Profile	Average Daily Consumption (kWh)	Average Peak Demand (kW)	Cluster Size and Percentage of Total Consumption (%)
I	High consumption: 80% Normal rate, 19% Economic rate; 88% detached house, 7% summer cottage.	63.0	5.1	10.0, 28.9
II	Medium to high consumption: 94% Normal rate, 5% Economic rate; 75% detached house, 18% summer cottage, 6% town house.	39.3	3.2	17.0, 30.7
III	Medium to low consumption: 96% Normal rate; 75% detached house, 9% summer cottage, 12% town house.	20.3	2.0	25.9, 24.1
IV	Low consumption: 99% Normal rate; 18% detached house, 70% summer cottage, 8% town house.	7.5	0.6	47.1, 16.2

Acknowledgements. The authors gratefully acknowledge the financial support of the Academy of Finland (grant nos. 127592 and 127656) and the Fortum Foundation. The case organization's cooperation is also gratefully acknowledged.

References

- Abdel-Aal, R.E.: Short Term Hourly Load Forecasting Using Abductive Networks. *IEEE Transactions on Power Systems* 19(1), 164–173 (2004)
- Back, B., Toivonen, J., Vanharanta, H., Visa, A.: Comparing numerical data and text information from annual reports using self-organizing maps. *International Journal of Accounting Information Systems* 2(4), 249–269 (2001)
- Baragoin, C., Andersen, C., Bayerl, S., Bent, G., Lee, J., Schommer, C.: *Mining Your Own Business in Retail Using DB2 Intelligent Miner for Data*. IBM Redbooks (2001)
- Berry, M.J.A., Linoff, G.: *Data mining techniques: for marketing, sales, and customer relationship management*. Wiley Computer Publishing (2004)
- Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford University Press, Avon (1995)
- CEER Advice on the take-off of a demand response electricity market with smart meters, Ref: C11-RMF-36-03 (December 2011)

- Charytoniuk, W., Chen, M.-S.: Very Short Term Load Forecasting Using Artificial Neural Networks. *IEEE Transactions on Power Systems* 15(1), 363–368 (2000)
- Collica, R.S.: *CRM Segmentation and Clustering Using SAS Enterprise Miner*. SAS Publishing (2007)
- Cotti, M., Millan, R.: Cervantes project and 'meters and more': the state of the art of smart metering implementation in Europe. In: *Proceeding of 21st International Conference on Electricity Distribution (CIRED 2011)*, Frankfurt, Germany, paper 0829 (2011)
- Deboeck, G., Kohonen, T.: *Visual explorations in finance using self-organizing maps*. Springer, Berlin (1998)
- Eklund, T., Back, B., Vanharanta, H., Visa, A.: Using the self-organizing map as a visualization tool in financial benchmarking. *Information Visualization* 2(3), 171–181 (2003)
- Garpetun, L.: Experiences from operations after a full-scale smart metering rollout regarding availability and reliability. In: *Proceedings of the 21st International Conference on Electricity Distribution (CIRED 2011)*, Frankfurt, Germany, paper 0415 (2011)
- Han, J., Kamber, M.: *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann (2000)
- Kaski, S., Kangas, J., Kohonen, T.: Bibliography of Self-Organizing Map (SOM) Papers 1981–1997. *Neural Computing Surveys* 1, 102–350 (1998)
- Keppo, J., Räsänen, M.: Pricing of electricity tariffs in competitive markets. *Energy Economics* 21, 213–223 (1999)
- Kohonen, T.: *Self-organizing maps*. Springer Series in Information Sciences, vol. 4, pp. 22–25. Springer, Berlin (1997)
- Kohonen, T.: *Self-organizing maps*, 3rd edn. Springer, Berlin (2001)
- Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J.: *SOM PAK: The self-organizing map program package*. Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science (1996)
- Lee, S.-C., Gu, J.-C., Suh, Y.-H.: A Comparative Analysis of Clustering Methodology and Application for Market Segmentation: K-Means, SOM and a Two-Level SOM. *Foundations of Intelligent Systems*, 435–444 (2006)
- Lendasse, A., Lee, J., Wertz, V., Verleysen, M.: Forecasting electricity consumption using nonlinear projection and self-organizing maps. *Neurocomputing* 48, 299–311 (2002)
- Liu, H., Eklund, T., Back, B., Vanharanta, H.: *Visual Data Mining: Using Self-Organizing Maps for Electricity Distribution Regulation*. In: Ariwa, E., El-Qawasmeh, E. (eds.) *DEIS 2011*. CCIS, vol. 194, pp. 631–645. Springer, Heidelberg (2011)
- Mutanen, A., Repo, S., Järventausta, P.: AMR in distribution state estimation. In: *Proceedings of Nordic Distribution and Asset Management Conference*, Bergen, Norway (2008)
- Mutanen, A., Repo, S., Järventausta, P.: Customer Classification and Load Profiling Based on AMR Measurements. In: *Proceedings of the 21st International Conference on Electricity Distribution (CIRED 2011)*, Frankfurt, Germany, paper 0277 (2010)
- Nababhushana, T.N., Veeramanju, K.T., Shivanna: Coherency identification using growing self organizing feature maps (power system stability). In: *IEEE Proceedings of EMPD 1998*. International Conference on Energy Management and Power Delivery, vol. 1, pp. 113–116 (1998)
- Oja, M., Kaski, S., Kohonen, T.: Bibliography of Self-Organizing Map (SOM) Papers: 1998–2001 Addendum. *Neural Computing Surveys* 3, 1–156 (2002)
- Rehtanz, C.: Visualisation of voltage stability in large electric power systems. In: *IEEE Proceedings Generation, Transmission and Distribution*, vol. 146, pp. 573–576 (1999)

- Riqueline, J., Martinez, J.L., Gomez, A., Goma, D.C.: Possibilities of artificial neural networks in short-term load forecasting. In: Proceedings of the IASTED International Conference Power and Energy Systems, pp. 165–170. IASTED/ACTA Press, Anaheim (2000)
- Sarlin, P., Peltonen, T.: Mapping the State of Financial Stability. ECB WP No. 1382/2011 (2011)
- Samarasinghe, S.: Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition. CRC Press (2007)
- Valtonen, P., Honkapuro, S., Partanen, J.: Improving Short-Term Load Forecasting Accuracy by Utilizing Smart Metering. In: Proceedings of the 20 th International Conference on Electricity Distribution (CIRED 2010), Lyon, France, paper 0056 (2010a)
- Vesanto, J., Alhoniemi, E.: Clustering of the self-organizing map. IEEE Transactions on Neural Networks 11(3), 586–600 (2000)
- Ward, J.H.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association 58(301), 236–244 (1963)
- Wiskott, L., Sejnowski, T.: Constrained optimization for neural map formation: A unifying framework for weight growth and normalization. Neural Computation 10(3), 671–716 (1998)
- Yao, Z., Holmbom, A.H., Eklund, T., Back, B.: Combining Unsupervised and Supervised Data Mining Techniques for Conducting Customer Portfolio Analysis. In: Perner, P. (ed.) ICDM 2010. LNCS, vol. 6171, pp. 292–307. Springer, Heidelberg (2010)

Wind Turbines Fault Diagnosis Using Ensemble Classifiers

Pedro Santos¹, Luisa F. Villa², Anfbal Reñones², Andrés Bustillo¹, and Jesús Maudes¹

¹ Department of Civil Engineering, University of Burgos

C/ Francisco de Vitoria s/n, 09006, Burgos, Spain

{psgonzalez, abustillo, jmaudes}@ubu.es

² CARTIF Foundation

Parque Tecnológico de Boecillo, 47151 Boecillo, Valladolid, Spain

{luivil, aniren}@cartif.es

Abstract. Fault diagnosis in machines that work under a wide range of speeds and loads is currently an active area of research. Wind turbines are one of the most recent examples of these machines in industry. Conventional vibration analysis applied to machines throughout their operation is of limited utility when the speed variation is too high. This work proposes an alternative methodology for fault diagnosis in machines: the combination of angular resampling techniques for vibration signal processing and the use of data mining techniques for the classification of the operational state of wind turbines. The methodology has been validated over a test-bed with a large variation of speeds and loads which simulates, on a smaller scale, the real conditions of wind turbines. Over this test-bed two of the most common typologies of faults in wind turbines have been generated: imbalance and misalignment. Several data mining techniques have been used to analyze the dataset obtained by order analysis, having previously processed signals with angular resampling technique. Specifically, the methods used are ensemble classifiers built with *Bagging*, *Adaboost*, *Geneneral Boosting Projection* and *Rotation Forest*; the best results having been achieved with *Adaboost* using C4.5 decision trees as base classifiers.

Keywords: fault diagnosis, wind turbines, ensemble classifiers, angular resampling.

1 Introduction

Vibration analysis has been studied and applied to rotating machinery for decades. It is widely accepted as one of the main fault diagnosis techniques in machine maintenance [1]. As the signal analysis technology has advanced and new sensors have been developed, fault diagnosis and maintenance of machines working under more severe conditions have become a new target area for experts. Examples of machines that work under variable conditions of load and speed are wind turbines, excavators and helicopters [2]; [4]; [5]; [3]. Gear transmission plays a crucial role in the reliability of these machines.

One of the first research in the field of transmission damage diagnosis focused on vibration signals analysis [6]. At first, the statistical features of the signal in the time

domain were the main element of study [18]. However, the field quickly spread to include spectral analysis, time-frequency analysis and, finally, models based on artificial intelligence. All of these approaches are still valid and current. As new techniques of signal processing arise, they are applied to the problem of damage detection in chain drives and should be adapted to the needs and specific characteristics of each mechanical system.

The main purpose of this work is to study fault diagnosis in wind turbines. To do so, the test-bed shown in Figure 1 is used to approximate real conditions and the typical faults of a real wind turbine.

Many studies have applied several signal analysis methods that are suited to conditions of fluctuating loads. Among these, we may quote works by Stander, Heyns, Zhan and Barlelmus [21]; [24]; [3]. However, no studies have yet been completed on such wide working ranges as those of a wind turbine, in terms of real wind regimes that therefore have a very wide range of speed and load operating conditions. The development of intelligent devices, both for monitoring and for diagnosis of this type of industrial equipment that operates under highly variable loads and speeds is, therefore, a highly topical field of research. Vibration monitoring systems require signal processing procedures to compensate for fluctuations in axis speeds and amplitude modulation, due to the variable wind-resistance loads [20]; [19].

Although exhaustive research into the analysis of the signals obtained from several types of sensors and particularly accelerometers has been completed to date, the standard technique used for fault diagnosis is the identification of critical variables by an expert and the development of a regression model that forecasts the failure [24]. The aim of this work is to develop an alternative classification system with greater reliability using ensemble classifiers.

There are several works in which ensemble classifiers have been used for fault detection. In Hu [12] *Adaboost* is used to combine *Support Vector Machines* (a type of base classifier) for fault diagnosis in rotating machinery. This method is also used in Donat [8] for the fault detection of engines in gas turbines. In Alonso [1], failure identification in continuous processes is managed by an ensemble classifier building method -*Stacking*- that combines nearest-neighbours base classifiers (*k-Neighbours Classifier*, *kNN*). Furthermore, *Adaboost* and *Bagging* of neural networks in El-Gamal [9] are used for fault diagnosis in analogue circuits.

2 Description of the Test-Bed and Measurement Procedure

The experiments conducted on the test-bed are meant to simulate the behaviour of wind turbines. This test-bed is used to simulate different defects under variable loads and speeds. The right side of the test-bed (Figure 1) is composed of an engine, a parallel gearbox and a planetary gearbox. Both gearboxes resemble a commercial wind turbine in terms of their configuration and gear ratios (1:61).

To simulate the variable load in the drive train of a wind turbine, due to randomness of the wind, an electric brake has been added to the right side of the bench.

For the measurement of vibration signals four accelerometers distributed in the axial and radial position in the gearboxes situated on the right side of the test-bed were used.

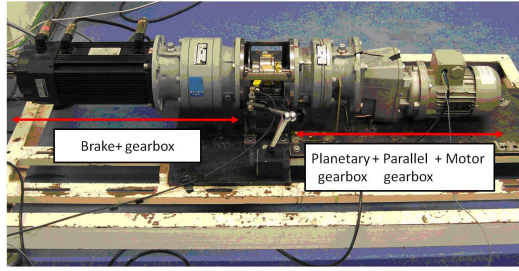


Fig. 1. Test-Bed

Preliminary processing of the vibration signals need to be performed, due to the speed and load variations caused by the operating conditions of wind turbines, in order to extract the information on its spectral analysis. The technique of angular sampling, a methodology that may be found in [22], appears suitable to solve this problem.

The faults simulated on the test-bed were imbalance and misalignment, starting with small values and increasing at each measurement to simulate a progressive fault (Table 1). This table illustrates the value of the weight in grams and its equivalent in percentages with regard to the total of the weight of the rotor of the bench, and the thickness used for producing the misalignment, as well as the resulting value.

Table 1. Types of faults and magnitudes induced in the test-bed

	Imbalance		Misalignment		
	gr	%		mm	°
Imbalance A	5.79	0.077	Misalignment A	0.75	1.53
Imbalance B	9.13	0.12	Misalignment B	0.75	1.53
Imbalance C	19.5	0.26			
Imbalance D	28.8	0.38			

To guarantee the speed and load conditions, several profiles were generated to cover a wide range of speeds from 1000 to 1800 rpm at random, and from 0 to 100 % of the load. An example of this profile is shown below in the Figure 2.

These profiles were generated to cover a whole day of measurement (24 hours), with constant 100 second intervals of speed and load. The speed measurements were generated from 1000 rpm, which is the approximate speed at which a wind turbine begins to produce energy. Data acquisition was taken at intervals of 72 seconds from each of the four accelerometers with a sampling frequency of 25600 Hz. The speed signal was captured at 6400 Hz.

The set of tests done are reported in [23].

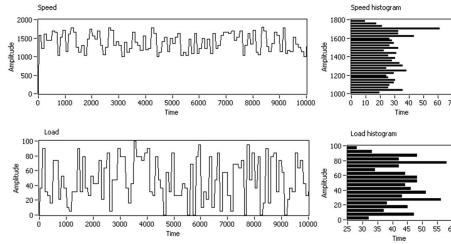


Fig. 2. Speed and load profile

3 Variables Analyzed

As explained in the previous section, several working faults in the turbine are analyzed. For that reason, a discrete output variable was defined, referred to as the fault type, and several input variables.

The type of fault matches the two previously explained ones; misalignment and imbalance, for which there are three possible numerical values in the first case (0, 0.75 and 2) and five in the second one (0, 5.79, 9.13, 19.5 and 28.8). We will refer to these degrees of misalignment as DA1, DA2 and DA3, and to imbalance as DB1, DB2, DB3, DB4, DB5.

There are therefore fifteen possible values for each type of faults, shown below in Table 2:

Table 2. Fault Classes

	DB1	DB2	DB3	DB4	DB5
DA1	0 (C0)	1 (C1)	2 (C2)	3 (C3)	4 (C4)
DA2	5 (C5)	6	7	8	9
DA3	10 (C6)	11	12	13	14 (C7)

In the previous table, class 0 matches the case in which there is no fault (no misalignment nor imbalance), and the 14 remaining classes match several types of faults that could theoretically occur, but in the experimental trials only 8 classes took place. These fault classes will be referred to as C0, C1, C2, C3, C4, C5, C6 and C7.

The variables in this problem are, on the one hand, 3 magnitudes which describe the operational state of the machine in the terms of torque, speed and electric input current and, on the other, several magnitudes measured with 5 sensors, 1 current sensor and 4 accelerometers, 2 by each of the two gearboxes, distributed along two perpendicular axis.

The current sensor provides 4 measurements of electric current, and the accelerometers provide the data for a vibration analysis along the axis, by using three aspects of the vibration spectrum. On the one hand, 5 measurements which summarize their distribution (average, RMS, skewness, kurtosis and interquartile range); on the other,

Table 3. Input variables

<i>Operation State</i>		
Variable	Number of measurments	Units
Torque	1	% of maximum torque
Speed	1	Hz
Input current	1	Amperes
<i>Current Sensors</i>		
Variable	Number of measurments	Units
Electrical current in the axis	4	Amperes
<i>Vibration analysis</i>		
Variable	Number of measurments	Units
Harmonics	272	mm/s^2
Bands	245	mm/s^2
Average	4	mm/s^2
RMS	4	mm/s^2
Skewness	4	dimensionless
Kurtosis	4	dimensionless
Interquartile Range	4	mm/s^2

a harmonic analysis (natural frequency of system vibrations and multiples thereof, 80 measurements in total); and finally, dividing the vibration spectrum into bands of fixed position (unrelated to the natural frequency of the system), with another 77 measurements. Each accelerometer provides a total of 162 measurements, although the total number of considered variables in the vibration analysis is 537, as some measurements with redundant information have been removed.

The final number of variables for the problem is 544, adding to the 537 from the vibration analysis, the measurements from the current sensor, the torque, the speed and the electric current. In the next table, a summary of the previously explained variables is completed, although it is possible to search for a more detailed information in [23].

During the day of the experimentation, 6551 different conditions in the considered variables were registered. The data set under study therefore has a size of 6551 instances with 544 attributes, such that it can be considered a high dimensional problem.

The distribution of the instances among the classes is as shown in Table 4:

Table 4. Distribution of the instances among the classes

C0	887 (13.54 %)
C1	847 (12.93 %)
C2	856 (13.07 %)
C3	838 (12.79 %)
C4	864 (13.19 %)
C5	872 (13.31 %)
C6	835 (12.75 %)
C7	552 (8.43 %)

4 Fault Analysis by Ensemble Classifiers

Forecasting several faults that may occur in turbine operation is included in data mining classification problems. In this article, the use of techniques to combine several individual classifiers is proposed, to obtain an ensemble classifier. These techniques have developed over the last decade and their output has been proven in several situations.

An ensemble classifier is a classification technique by which the forecasted class is obtained from the individual forecasts of a series of base classifiers. There are several ways of combining the various forecasts, the most usual one is to select the most voted class. The global accuracy of the ensemble classifier depends on the diversity of the classifiers and their individual accuracy, as an ensemble classifier should be capable outperforming any individual classifier [7]; [14].

There are several ways of forcing diversity between base classifiers [13]; [17], having taken four of these techniques in this study, *Bagging* and *Adaboost* on the one hand, are the most commonly used, and *Rotation Forest* and *General Boosting Projection (GBPC)* on the other, which are more novel techniques that have been shown to be very competitive [16]; [10].

The algorithm *Rotation Forest* algorithm is based on Principal Component Analysis (PCA) extraction procedures that achieve better accuracy in the ensemble classifier, by acting at the same time on the individual accuracy of each base classifier and on its diversity [16]. Thus, a random division of the data is made, in groups of attributes (3 in this work), and subsequently a PCA analysis is completed over part of the samples of each group, also random, storing the projection matrix that is used and combined later on to project all the samples of each group.

The *GBPC (General Boosting Projection)* is based on the use of supervised projections to improve global accuracy, due to the individual improvement of each base classifier as well as its diversity [10]. It is an iterative process in which the first base classifier receives the data set without any modification followed by a projection over the misclassified instances by the previous classifier. By doing so, we seek to obtain better results in the next classifier, in cases where the previous classifiers failed. The Non-parametric Discriminant Analysis (NDA) version proposed by [15] was used as the supervised projection method.

5 Results

Three methodologies for the classification were tested: C4.5 decision trees, k-Nearest Neighbour (*kNN*) and Naive Bayes. These three base techniques were chosen as they are the three most commonly used in data mining.

These methods have been tested individually as well as with ensemble classifiers using the techniques of *Bagging*, *Adaboost*, *GBPC* and *Rotation Forest*, taking in all cases 100 base classifiers, and performing a 5×2 cross validation (all the methods are compared using the same sets for training and testing).

Two ways to measure the accuracy of each classifier have been taken:

- Success rate in a 5×2 cross validation, indicating the standard deviation of the iterations.
- Confusion matrix, in which the class forecasted by the classifier is compared against the class of the instance to which actually belongs.

5.1 Success Rate

The following table illustrates the success rate of both the individual and the different ensemble classifiers, which includes the standard deviation with regard to the 5 repetitions of the cross validation between parentheses.

In all cases, we can see that decision trees are more suitable as base classifiers, however we should highlight the notable increase of the *GBPC* with regard to the efficiency of the classification with the *kNN* as base classifier.

Table 5. Average success and standard deviation for the different classifiers

	C4.5 trees	kNN	Naive Bayes
Classifier individually	92.60 (0.51)	66.12 (0.44)	70.29 (2.68)
Bagging	95.33 (0.23)	66.01 (0.41)	70.73 (1.59)
Adaboost	96.24 (0.12)	67.57 (0.59)	78.96 (0.71)
GBPC (NDA)	90.70 (5.61)	87.45 (1.26)	70.29 (2.68)
Rotation forest	95.84 (0.14)	66.19 (0.54)	71.92 (2.25)

The low performance of the *kNN* classifier could be caused by the well-known problem of the "curse of dimensionality" (analyzing high-dimensional spaces). In the following sections we compare the two methods in which better results are reached, *Adaboost* with decision trees versus *Rotation Forest* with decision trees.

5.2 Confusion Matrix

The next step is to compare the results of *Adaboost* and *Rotation Forest* with 100 C4.5 trees as base classifiers, by using the average confusion matrix of the 5×2 cross validation (the confusion matrix average of those provided by each of the 10 classifiers obtained in the cross validation has been calculated, and the values have been rescaled with regard to the total).

Regarding to the operation control, the most critical cases are those registered in the first column in both tables, as they match with those in which the ensemble classifier estimates that there are not a fault operation. By analyzing the data of this column, we can see that the undetected percentage of errors is 0.23 % in the case of *Adaboost*, and 0.68 % in the case of *Rotation Forest*.

Using the *t* test to compare the ensemble classifiers *Adaboost* and *Rotation forest* with a level of significance of 1 %, we may conclude from the statistical evidence that the first algorithm outperforms the second one with regard to the way it models the data.

Table 6. Confusion matrix for *Adaboost* (top) / *Rotation Forest* (bottom) of C4.5 trees

	C0	C1	C2	C3	C4	C5	C6	C7
C0	13.40	0.00	0.14	0.00	0.00	0.00	0.00	0.00
C1	0.00	11.53	0.03	1.18	0.19	0.00	0.00	0.00
C2	0.01	0.00	12.87	0.19	0.00	0.00	0.00	0.00
C3	0.02	0.56	0.27	11.23	0.71	0.00	0.00	0.00
C4	0.01	0.04	0.00	0.41	12.73	0.00	0.00	0.00
C5	0.00	0.00	0.00	0.00	0.00	13.31	0.00	0.00
C6	0.00	0.00	0.00	0.00	0.00	0.00	12.75	0.00
C7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8.42

	C0	C1	C2	C3	C4	C5	C6	C7
C0	13.39	0.00	0.15	0.00	0.00	0.00	0.00	0.00
C1	0.00	11.80	0.03	0.90	0.20	0.00	0.00	0.00
C2	0.03	0.00	12.88	0.15	0.00	0.00	0.00	0.00
C3	0.02	0.78	0.48	10.72	0.79	0.00	0.00	0.00
C4	0.04	0.09	0.02	0.46	12.58	0.00	0.00	0.00
C5	0.00	0.00	0.00	0.00	0.00	13.31	0.00	0.00
C6	0.00	0.00	0.00	0.00	0.00	0.02	12.73	0.00
C7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8.43

6 Conclusions

This study has proposed a fault diagnosis system for machines with high variation in the speed and load conditions, such as wind turbines. These devices have undergone significant growth over the last five years and require immediate industrial solutions to their tele-maintenance problems. The failure diagnosis system explained in this work consists of several measurement sensors, especially accelerometers, signal analysis equipment based on resampling angular techniques to process the data from these sensors, and a module that implements different data mining techniques for the classification of the operational state of wind turbines. Several methods of combining base classifiers have been applied to identify seven different levels of two typical faults in wind turbines: misalignment and imbalance. *Adaboost* using J48 decision trees as base classifiers achieved high accuracy (correct forecasts in 96.24 % of cases) when analyzing a wide real dataset measured on a test-bed that simulate real conditions of wind turbines operation (65551 instances with 544 attributes). Future research will be focused in the improvement of the industrial application through the testing of the proposed fault diagnosis system on a more extensive dataset that includes more fault cases and has been recorded under real industrial conditions, because the analysed dataset reflects a limited number cases of two fault types (misalignment and imbalance).

Acknowledgments. This work has been partially funded by the Ministry of Science and Innovation of Spain through the MAGNO project (Ref. 2008/1028), within the CENIT funding programme.

References

1. Alonso, C.J., Prieto, O.J., Rodríguez, J.J., Bregón, A., Pulido, B.: Stacking Dynamic Time Warping for the Diagnosis of Dynamic Systems. In: Borrajo, D., Castillo, L., Corchado, J.M. (eds.) CAEPIA 2007. LNCS (LNAI), vol. 4788, pp. 11–20. Springer, Heidelberg (2007), http://dx.doi.org/10.1007/978-3-540-75271-4_2
2. Barszcz, T., Randall, R.B.: Application of spectral kurtosis for detection of a tooth crack in the planetary gear of a wind turbine. *Mechanical Systems and Signal Processing* 23(4), 1352–1365 (2009), <http://www.sciencedirect.com/science/article/pii/S0888327008002239>
3. Bartelmus, W., Zimroz, R.: Vibration condition monitoring of planetary gearbox under varying external load. *Mechanical Systems and Signal Processing* 23(1), 246–257 (2009), <http://www.sciencedirect.com/science/article/pii/S0888327008000824>, special Issue: Non-linear Structural Dynamics
4. Blunt, D.M., Keller, J.A.: Detection of a fatigue crack in a uh-60a planet gear carrier using vibration analysis. *Mechanical Systems and Signal Processing* 20(8), 2095–2111 (2006), <http://www.sciencedirect.com/science/article/pii/S0888327006001245>
5. Combet, F., Zimroz, R.: A new method for the estimation of the instantaneous speed relative fluctuation in a vibration signal based on the short time scale transform. *Mechanical Systems and Signal Processing* 23(4), 1382–1397 (2009)
6. Davies, A.: *Handbook of condition monitoring: techniques and methodology*. Chapman & Hall (1998), <http://books.google.es/books?id=j2mW2aIs2YIC>
7. Dietterich, T.: Ensemble Methods in Machine Learning. In: MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000), http://dx.doi.org/10.1007/3-540-45014-9/_1, 10.1007, doi:10.1007/3-540-45014-9_1
8. Donat, W., Choi, K., An, W., Singh, S., Pattipati, K.: Data visualization, data reduction and classifier fusion for intelligent fault diagnosis in gas turbine engines. *Journal of Engineering for Gas Turbines and Power* 130(4), 041602 (2008), <http://link.aip.org/link/?GTP/130/041602/1>
9. El-Gamal, M., Mohamed, M.: Ensembles of neural networks for fault diagnosis in analog circuits. *Journal of Electronic Testing* 23, 323–339 (2007), <http://dx.doi.org/10.1007/s10836-006-0710-1>, doi:10.1007/s10836-006-0710-1
10. García-Pedrajas, N., García-Osorio, C.: Constructing ensembles of classifiers using supervised projection methods based on misclassified instances. *Expert Syst. Appl.* 38(1), 343–359 (2011)
11. Hameed, Z., Hong, Y., Cho, Y., Ahn, S., Song, C.: Condition monitoring and fault detection of wind turbines and related algorithms: A review. *Renewable and Sustainable energy reviews* 13(1), 1–39 (2009)
12. Hu, Q., He, Z., Zhang, Z., Zi, Y.: Fault diagnosis of rotating machinery based on improved wavelet package transform and svms ensemble. *Mechanical Systems and Signal Processing* 21(2), 688–705 (2007), <http://www.sciencedirect.com/science/article/pii/S0888327006000306>
13. Kuncheva, L.: Combining classifiers: Soft computing solutions. *Pattern Recognition: From Classical to Modern Approaches*, 427–451 (2001)
14. Kuncheva, L.: *Combining pattern classifiers: methods and algorithms*. Wiley-Interscience (2004), <http://books.google.es/books?id=9TJ6iGZtqWAC>
15. Kuo, B., Ko, L., Pai, C., Landgrebe, D.: Regularized feature extractions for hyperspectral data classification. In: 2003 IEEE International Proceedings of Geoscience and Remote Sensing Symposium, IGARSS 2003, vol. 3, pp. 1767–1769. IEEE (2003)

16. Rodriguez, J., Kuncheva, L., Alonso, C.: Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(10), 1619–1630 (2006)
17. Rokach, L.: Ensemble-based classifiers. *Artificial Intelligence Review* 33, 1–39 (2010), <http://dx.doi.org/10.1007/s10462-009-9124-7>, doi:10.1007/s10462-009-9124-7
18. Samuel, P.D., Pines, D.J.: A review of vibration-based techniques for helicopter transmission diagnostics. *Journal of Sound and Vibration* 282(1-2), 475–508 (2005), <http://www.sciencedirect.com/science/article/pii/S0022460X04003244>
19. Stander, C.J., Heyns, P.S.: Instantaneous angular speed monitoring of gearboxes under non-cyclic stationary load conditions. *Mechanical Systems and Signal Processing* 19(4), 817–835 (2005), <http://www.sciencedirect.com/science/article/pii/S0888327004001633>
20. Stander, C.J., Heyns, P., Schoombie, W.: Using vibration monitoring for local fault detection on gears operating under fluctuating load conditions. *Mechanical Systems and Signal Processing* 16(6), 1005–1024 (2002), <http://www.sciencedirect.com/science/article/pii/S0888327002914792>
21. Stander, C., Heyns, P.: Transmission path phase compensation for gear monitoring under fluctuating load conditions. *Mechanical Systems and Signal Processing* 20(7), 1511–1522 (2006), <http://www.sciencedirect.com/science/article/pii/S0888327005000919>
22. Villa, L.F., Renones, A., Perán, J.R., de Miguel, L.J.: Angular resampling for vibration analysis in wind turbines under non-linear speed fluctuation. *Mechanical Systems and Signal Processing* 25(6), 2157–2168 (2011), <http://www.sciencedirect.com/science/article/pii/S0888327011000677>, interdisciplinary Aspects of Vehicle Dynamics
23. Villa, L.F., Renones, A., Perán, J.R., de Miguel, L.J.: Statistical fault diagnosis based on vibration analysis for gear test-bench under non-stationary conditions of speed and load. In: *Mechanical Systems and Signal Processing* (in Press, 2012) a(accepted manuscript), doi:10.1016/j.ymssp.2011.12.013
24. Zhan, Y., Makis, V., Jardine, A.K.: Adaptive state detection of gearboxes under varying load conditions based on parametric modelling. *Mechanical Systems and Signal Processing* 20(1), 188–221 (2006), <http://www.sciencedirect.com/science/article/pii/S0888327004001499>

Bus Bunching Detection by Mining Sequences of Headway Deviations

Luís Moreira-Matias^{1,2}, Carlos Ferreira^{2,3}, João Gama^{2,5},
João Mendes-Moreira^{1,2}, and Jorge Freire de Sousa⁴

¹Departamento de Engenharia Informática, Faculdade de Engenharia,
Universidade do Porto, Rua Dr. Roberto Frias, s/n 4200-465 Porto – Portugal

²LIAAD-INESC Porto L.A. Rua de Ceuta, 118, 6º; 4050-190 Porto – Portugal

³Instituto Superior de Engenharia do Porto, Instituto Politécnico do Porto,
Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto

⁴Departamento de Engenharia Industrial e Gestão, Faculdade de Engenharia,
Universidade do Porto, Rua Dr. Roberto Frias, s/n 4200-465 Porto – Portugal

⁵Faculdade de Economia, Universidade do Porto

Rua Dr. Roberto Frias, s/n 4200-465 Porto – Portugal

{luis.matias, jmoreira, jfsousa}@fe.up.pt,

cgf@isep.ipp.pt, jgama@fep.up.pt

Abstract. In highly populated urban zones, it is common to notice headway deviations (HD) between pairs of buses. When these events occur in a bus stop, they often cause bus bunching (BB) in the following bus stops. Several proposals have been suggested to mitigate this problem. In this paper, we propose to find BBS (Bunching Black Spots) – sequences of bus stops where systematic HD events cause the formation of BB. We run a sequence mining algorithm, named PrefixSpan, to find interesting events available in time series. We prove that we can accurately model the BB trip usual pattern like a frequent sequence mining problem. The subsequences proved to be a promising way of identify the route’ schedule points to adjust in order to mitigate such events.

Keywords: Sequence Mining, Bus Bunching, Headway Irregularities.

1 Introduction

In highly populated urban zones, it is well known that there is some schedule instability, especially in highly frequent routes (10 minutes or less) [1-5]. In this kind of routes it is more important the headway (time separation between vehicle arrivals or departures) regularity than the fulfillment of the arrival time at the bus stops [4]. Due to this high frequency, this kind of situations may force a bus platoon running over the same route. In fact, a small delay of a bus provokes the raising of the number of passengers in the next stop. This number increases the dwell time (time period where the bus is stopped at a bus stop) and obviously also increases the bus’s delay. On the other hand, the next bus will have fewer passengers, shorter dwell times with

no delays. This will continue as a snow ball effect and, at a further point of that route, the two buses will meet at a bus stop, forming a platoon as it is illustrated in Fig. 1. This phenomenon has several denominations: the Bangkok effect [6], Bus Platooning [7], Vehicle Pairing [8], Headway Instability [1], Bus Clumping or Bus Bunching (BB) [9], [2]. From now on, we will use the last one.

The occurrence of BB forces the controllers to take actions in order to avoid this headway instability, forcing the adherence to the schedule. BB situations can cause several problems like: further buses delays, full buses, decreased comfort in the buses, larger waiting times at the bus stops, growing number of passengers waiting, greater resources demand and a decrease of schedule reliability. All this can cause the loss of passengers to other transportation means and/or companies.

Our goal is to identify the causes of BB occurrences using AVL (Automatic Vehicle Location) historical data. The BB phenomenon always starts by a headway deviation (HD) at a bus stop [10]. We intend to find frequent and systematic HD event sequences in the trips of a given route: bus stops where the bus activities - like the passenger boarding - will propagate the headway irregularities further and further. These bus stops sequences **highlights problematic route regions**: from now on we will refer to it as **Bunching Black Spots** (BBS - bus stops sequences where a HD will, with a high probability, start a BB in one of the following bus stops of the trip).

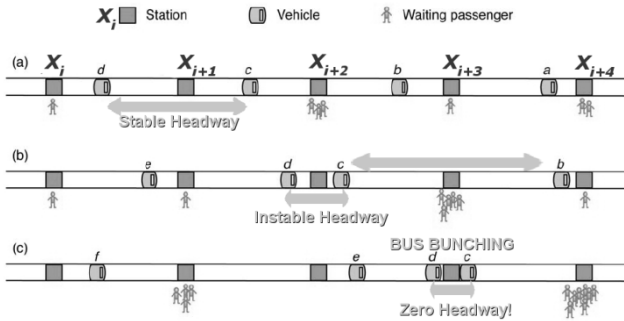


Fig. 1. Bus Bunching problem illustration. Figure based on Fig.1 from [1].

We use the PrefixSpan algorithm (presented in Section 3) to mine frequent sequences in the HD sequences extracted from this dataset. We apply this methodology to data from two urban lines of a public transport operator of Porto. It proved to be efficient in the detection of HD patterns in the bus stops of the studied routes.

The results from this framework can be highly useful to the public transport planners. One of the most known ways to mitigate the bus bunching is to adjust the slack time introduced in each schedule point (bus stops selected along the route for which the arrival time is defined) [11]. By using this framework, the planners can use the information about the BBS along the routes to select which schedule points should be changed (increasing or decreasing the slack time) to mitigate BB effectively.

The main results are: the observation that the BB phenomenon starts at the initial bus stops; and the existence of high correlation between HD that occurs at a given bus stop and the HD detected in the next ones.

This paper is structured as follows. Section 2 states a brief description of the problem we want to solve, the related work, our motivation and a clear definition of our approach. Section 3 presents the methodology proposed. Section 4 presents summarily the dataset used, its main characteristics and some statistics about it. Section 5 presents the results obtained through the application of the PrefixSpan algorithm to our dataset and a discussion about those results. Section 6 concludes and describes the future work we intend to carry on.

2 Problem Overview

Nowadays, the road public transportation (PT) companies face a huge competition of other companies or even of other transportation means like the trains, the light trams or the private ones. The service reliability is a fundamental metric to win this *race* [12]: if a passenger *knows* that a bus of a selected company will arrive *certainly* on the schedule on his bus stop, he will probably pick it often. The reverse effect is also demonstrated and a BB event forming a visual bus pair is a strong bad reliability signal to the passengers' perception of the service quality, which can lead to important profit losses [9, 13]. This tendency to form platoons is usual for urban vehicles (specially the PT ones) and arises for the specific and complex characteristics of transit service perturbations. Those are mainly related with changes in three key factors [8]: the dwell time and the loading time (highly correlated) and the non-casual passenger arriving (passengers that, for an unexpected reason – like a soccer match or a local holiday - try to board in a specific bus stop distinct from the usual one). However, the study of these changes impact on the service reliability is not in our current scope. Our goal is to find persistent and frequent headway irregularities which will *probably* provoke, in a short time horizon, a BB event.

There are two distinct approaches found in the literature to handle the BB events: the first one defines the bunching problem as a secondary effect of a traffic system malfunction like a traffic/logistic problem (signal priority handling, adaptation of bus stops/hubs logistics to the needs, adjustments of the bus routes to the passengers demand, etc.). The second one defines the BB problem like a main one that must be treated and solved *per se* (adjust the timetables and the schedule plans to improve schedules' reliability or set live actions to the irregular bus pairs, for instance).

In this work, we are just focused on the second approach which related work, motivation and scope we present along this section.

2.1 Related Work

There are two distinct approaches to mitigate BB: (1) the PT planning one, where they try to adjust the schedule plans somehow and the control one, where the BB is avoided by actions suggested live by the controllers and (2) the real-time approaches, which use

streaming data to evaluate the network and to choose some actions to keep the system stable. To do so, it is suggested one or more actions to the irregular (i.e. schedule behind or ahead) buses. There are four types of actions that can be proposed to avoid BB in real time: the change in bus holding time, the stop-skipping, the preplanning deadheading (the scheduling of some vehicles to run empty through a number of stations at the beginning or the end of their routes) and the change in the bus cruise speed.

We can split the existing experimental setups to test and evaluate such approaches in two big groups: the first one uses simulation models and the newer one's uses AVL historical data to test their approaches. A brief state-of-art on both is presented below.

Simulation Models

Newell *et. al* presented one of the first known models to reduce BB [14]: an optimization framework to control the headway deviation effects. Basically, it consists in the simulation of two buses and one control point. The simulation was run assuming ideal conditions and it consists in the introduction of delay in one of the buses using stochastic variables. The simulation tested control metrics to force the headway to remain stable.

Public transportation companies use slack times in the building of their schedule plans in order to avoid that delays in a given trip force delays in the departure of the next trip. This is a common practice in order to guarantee passengers' satisfaction by increasing schedules reliability. An important definition is presented by Zhao *et al.* in [11]: "*an optimal slack time will correspond to the best schedule plan possible. This plan should avoid BB situations*". They present a method to obtain the optimal slack times for a given number of vehicles on highly frequent routes.

One of the first probabilistic model to predict BB [15] defines a distribution along a given line to evaluate the tendency of buses to form pairs as they progress down their route. Other works present models like this one. One of them [16] uses the Monte Carlo theorem to introduce stochastic variations to the traffic conditions, namely, the bus speed between stops. Usually these works consider classical variables of public transportation planning like the bus speed between bus stops, passengers boarding time, headway, among others, to suggest forced actions to detect BB in a simulation. These two works suggest one or two types of forced actions to maintain stability in the simulation after the launch of a BB trigger.

Gershenson *et. al.* presented a model adapted from a metro-like system and implemented a multi-agent simulation [1]. To achieve stability, they implemented adaptive strategies where the parameters are decided by the system itself, depending on the passenger density. As a result, the system puts a restriction to the vehicle holding time (it sets a maximum dwell time), negotiating this value for each bus stop with the other vehicles.

Real Data (AVL) Models

The introduction of AVL systems changed the research point-of-view on bus bunching, in the last ten years, from planning to control. There are several techniques

in PT to improve the schedule plans on time tables based on AVL data. An useful review on those is presented by Peter Furth in [17].

C. Daganzo presents a dynamic holding time formulae based on real time AVL data in order to adaptively compensate the headway instability introduced in the system [2].

There are as well bus cruising speed approaches. In [3] it is presented a model allowing the buses to negotiate an ideal cruising speed to avoid potential BB situations.

Headway Irregularities on AVL-Based Models

The relations between the irregularities in the headway sequences and the BB events have been recently explored: in [8] is presented a study identifying the headway distributions representing service perturbations based on probability density functions (p.d.f.). This study was done using a stochastic simulation model for a one-way transit line accounting several characteristics like the dwell time or the arrivals during the dwell time (which values for each bus stops were calculated using the pre-calculated p.d.f.). Despite their useful conclusions, their model had two main disadvantages: 1) is not based in real AVL data and 2) it does not present a probability density function to represent the pattern of consecutive headways irregularities. We do believe that this specific issue can be rather addressed mining frequent sequences on real AVL data, as we present here.

2.2 Motivation and Scope

We can define the headway irregularities as events that occur in a bus stop of a given trip. Those events consist in a large variation (1 for positive or -1 for negative) on the headway: Headway Deviation events (HD).

These are usually correlated in a snowball effect that may occur (or not) in a given (straight or spaced) sequence of bus stops. Despite the analysis of the state-of-art work on the mitigation of BB events, the authors found no work on systematizing real HD patterns that seem to be in the genesis of a BB event.

An unreliable timetable is one of the main causes of many HD events. Usually, a timetable is defined using schedule points: stops for which there is an arriving or departing time defined. One of the most well-known PT planning ways to mitigate HD events is to add/reduce slack time in these defined timestamps to increase schedule plan overall reliability. However, only a small percentage of the bus stops served by a given timetable are used as schedule points. This is exemplified in the upper part of Fig. 2 (the reader can obtain further details on schedule plan building in chapter 1 from [18]). Usually, PT planners easily identify which lines present more HD and BB events. However, three questions still remain open:

- 1) Which should be the schedule points affected?
- 2) Which action (increase/decrease slack time) should be applied to these schedule points in order to reduce the occurrence probability of BB events?

- 3) Which day periods should have the timestamps in these schedule points changed?

In this work, we address the first and third questions by mining frequent HD event sequences in the trips of a given route: bus stops that systematically propagate the headway irregularities further and further. The second issue is out of our scope but it is well addressed in the literature [11].

Our intention is to point out a route region where an HD event fast and systematically propagates itself along the route, forming a Bunching Black Spot (BBS). The BBS can be specific of a period of the day or continuous along the day. In the bottom part of Fig. 2 we present an example of a BBS. In the next section we present our methodology to mine BBS.

3 Methodology

Our methodology consists in finding consistent patterns of frequent HD events occurring in the same bus stops whenever a BB occurs – BBS. To do so we compare, at each bus stop, the round-trip times of every consecutive bus pairs. With the HD series thus obtained, we mine frequent sequence patterns. Firstly, we introduce the algorithm we used and finally we describe how we use it to create and mine our HD series for a given route.

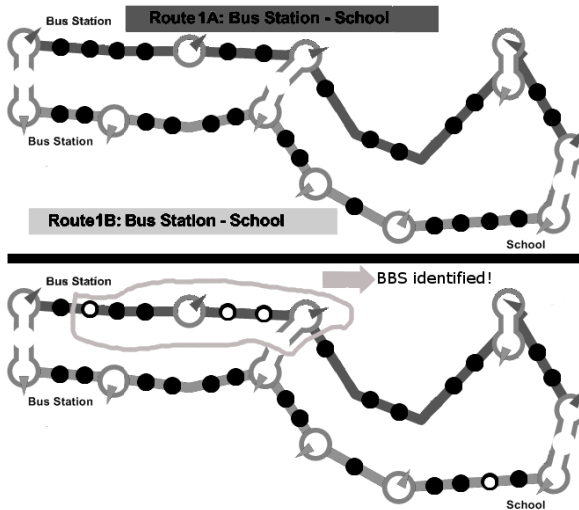


Fig. 2. Example of Schedule Points and BBS. The two schemas exemplify two routes of a line running between an arbitrary school and a main bus station. In top part, route 1A has 19 bus stops represented by 13 small black circles and 6 big grey circles (the single one's are just bus stops, the double are hubs/interfaces). The last ones are the schedule points in the route's timetables. In the bottom part, the stops belonging to frequent HD sequences are identified (even if the BB itself occurs later in the route) with a small white circle inside them. The highlighted stops form a route region (Bunching Black Spot) where the schedule points need to be time-adjusted.

3.1 Mining Time Series Sequences

There is a wide range of algorithms that can explore sequential data efficiently. To the best of our knowledge, Agrawal and Srikant introduced the sequential data mining problem in [19]. Let $\mathbf{I} = \{\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_n\}$ be a set of items and \mathbf{e} an event such that $\mathbf{e} \subseteq \mathbf{I}$. A sequence is an ordered list of events $\mathbf{e}_1\mathbf{e}_2\dots\mathbf{e}_m$ where each $\mathbf{e}_i \subseteq \mathbf{I}$.

Given two sequences $\alpha = \mathbf{a}_1\mathbf{a}_2\dots\mathbf{a}_r$ and $\beta = \mathbf{b}_1 \mathbf{b}_2 \dots \mathbf{b}_s$, sequence α is called a subsequence of β if there exists integers $1 \leq \mathbf{j}_1 < \mathbf{j}_2 < \dots < \mathbf{j}_r \leq s$ such that $\mathbf{a}_1 \subseteq \mathbf{b}_{\mathbf{j}_1}$, $\mathbf{a}_2 \subseteq \mathbf{b}_{\mathbf{j}_2}$, ... , $\mathbf{a}_r \subseteq \mathbf{b}_{\mathbf{j}_r}$. A sequence database is a set of tuples (\mathbf{sid}, α) where \mathbf{sid} is the sequence identification and α is a sequence. The count of a sequence α in \mathbf{D} , denoted $\mathbf{count}(\alpha, \mathbf{D})$, is the number of sequences in \mathbf{D} containing the α subsequence.

The support of a sequence α is the ratio between $\mathbf{count}(\alpha, \mathbf{D})$ and the number of sequences in \mathbf{D} . We denote sequence support as $\mathbf{support}(\alpha, \mathbf{D})$. Given a sequence database \mathbf{D} and a minimum support value λ , the problem of sequence mining is to find all subsequences in \mathbf{D} having a support value equal or higher than the λ value. Each one of the obtained sequences is also known as a frequent sequence.

In [20] the GSP algorithm, an algorithm that generalizes the original sequential pattern mining problem, is introduced. The search procedure of this algorithm is inspired by the well-known APRIORI algorithm [21]. GSP uses a candidate-generation strategy to find all frequent sequences, and uses a lattice to generate all candidate sequences. We observe that GSP has limitations when dealing with large datasets because candidate generation may require multiple database queries.

Several approaches have been proposed to address the above mentioned issue. One of the most interesting and efficient proposals is PrefixSpan algorithm [22]. This algorithm makes use of pattern-growth strategies to efficiently find the complete set of frequent sequences. The algorithm starts by finding all frequent items (length one sequences). Then, for each one of these frequent items (the prefix) PrefixSpan partitions the current database into *prefix projections*. Each projection database contains all the sequences with the given prefix. This procedure runs recursively until all frequent sequences are found.

In this work we run PrefixSpan algorithm to solve our problem due to its popularity and efficiency.

3.2 Methodology

Firstly we constructed headway sequences based in the AVL historic data for every bus pairs in a given route. Then we identified the headway profiles where BB events occurred based on the bus service reliability metrics presented in [23] and we extracted HD sequences from them.

Let $X = x_1x_2\dots x_n$ be a headway sequence measured between a bus pair in a given route through n bus stops running with a frequency f ($f = 1/x_1$). We identify a BB if there exists a x_i satisfying the inequality $x_i \leq (0.25 * 1/f)$ for at least one $i \in \{1, \dots, n\}$. An example of this analysis is shown in Fig. 3 and in Fig. 4, where we identified 4 BB events. Based on this headway profiles, we formed a HD sequence as follows. Let $H = h_1h_2\dots h_n$ be the HD sequences based on X . We compute the value of

each h_i (the headway between a bus pair in the bus stop x_i), for each $i \in \{2, \dots, n\}$, using the expression 1.

$$h_i = \begin{cases} 0 & \text{if } |x_i - x_{i-1}| < \left(\frac{1}{f}\right) * ht \\ 1 & \text{if } x_i - x_{i-1} \geq \left(\frac{1}{f}\right) * ht \\ -1 & \text{if } x_i - x_{i-1} \leq -\left(\frac{1}{f}\right) * ht \end{cases} \quad (1)$$

where ht is a threshold parameter given by the user for the HD definition. For the first bus stop is considered an HD of 0. Basically, a -1 event corresponds to a negative HD (delay) in a bus stop (i.e.: the two buses become closer), the 1 event is a positive HD (ahead of schedule) and the 0 occurs when the headway remains stable.

The x_n represents a headway deviation in a bus stop n . The HD sequences are ordered according to the bus stop order defined for a given route. Our goal is to find sequences of bus stops with frequent HD by exploring a set of trips, in a given route, where BB occurrences were identified.

To do so, we collected the HD sequences of trips in work days where a BB event occurred and we mined them using the PrefixSpan algorithm by setting a (user-defined) minimum support value in order to identify HD patterns in the bus stops. Fig. 5 illustrates our methodology. We applied this methodology to four routes in a given period. This data is summarily described in Section 4.

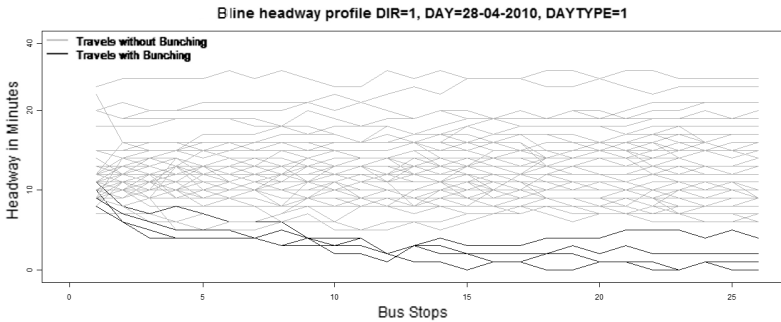


Fig. 3. Headway profiles of the route B1 for a given day. There were four BB events identified.

4 Dataset

The source of this data was STCP, the Public Transport Operator of Porto, Portugal. The dataset was obtained through a bus dispatch system that integrates an Automatic Vehicle Location (AVL) system. The data captured through this system contains data of the trips from two lines (A and B) in the working days for the first ten months of 2010. Each line has two routes – one for each way {A1, A2, B1, B2}. Line B is a

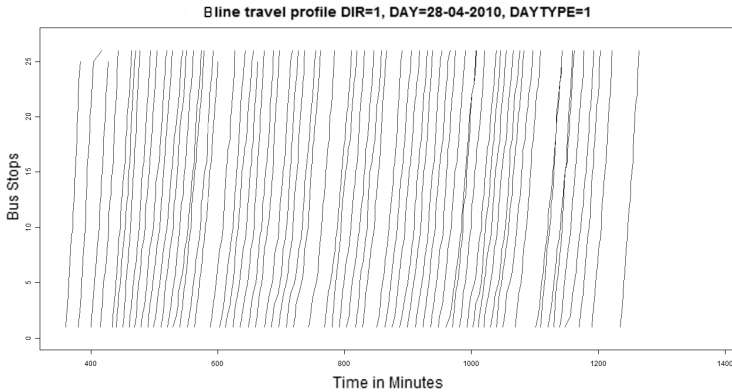


Fig. 4. Travel time profiles for the same day of Fig. 3. It is possible to identify the bunching situations directly.

common urban line between *Viso* (an important neighborhood in Porto) passing by 26 bus stops (BS1_B1 to BS26_B1 and BS1_B2 to BS26_B2, respectively), and ending at *Sá da Bandeira*, a downtown bus hub. Line A is also an urban line between another downtown bus hub (*Cordoaria*) and *Hospital São João* - an important bus/light train interface in the city – using 22 bus stops (same schema than line B). This dataset has one entry for each stop made by a bus running in the route during that period. It has associated a timestamp and a day type (1 for work days, 2-6 for other day types i.e.: holidays and weekends). Table 1 presents some statistics about the set of trips per route considered and the BB events identified. The *Nr. of Trips* is the total number of trips considered in the given route, *TT* is the round-trip time, expressed in minutes, and *DT* is the number of daily trips occurred. Finally, trips with BB are the trips where at least one BB situation occurs and HD events are the positive or negative events ($h_i = 1$ or $h_i = -1$, respectively) measured in every bus stops along every trip for a given line.

Table 1. Descriptive statistics for each route considered. These times are in minutes. TT means round-trip times. DT means daily trips. Based in our HD event definition, the maximum number of events for a time period is given as *Nr. of Bus Stops * Nr. Of Trips*.

	B1	B2	A1	A2
Nr. of Trips	9391	10675	13802	12753
Nr. of Bus Stops	26	26	22	22
Minimum TT	11	11	11	11
Maximum TT	78	82	70	65
Minimum of DT	39	39	33	36
Maximum of DT	74	74	89	88
Median TT	29	21	21	38
Nr. of Bus Stops	26	26	22	22
Nr. of Trips w/ BB	332	378	559	630
Nr. of HD events detected	26905	29911	42803	43525

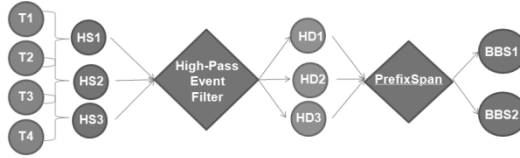


Fig. 5. Bunching Black Spot Detection Methodology illustration. T_n is the time series measured in each bus stop of a given trip. HS are the corresponding Headway Sequences and HD the Headway Deviation event subsequences.

5 Results

We did our experiments only for the trips occurred during the peak periods (08:00 to 11:00 and 16:00 to 19:00). We did so because BB mainly occurred – as expected – during those periods, as can be seen in Fig. 6. The routes A1 and A2 *suffer* more BB events and they are time-dispersed along the day. This happens because this line is an urban one between two important bus/metro interfaces (the downtown and the University Campus) with regular high frequencies during the entire day. So, they are highly frequent routes with many passengers during the entire day, which are well known factors to provoke BB occurrences. We mined sequences just in the bunching partition (trips with BB events). Moreover, we use the two partitions to compute the confidence of each sequence to be specific on the BB one. Our goal was to find patterns (i.e. frequent HD sequences) describing the headway irregular behavior of a typical BB trip in a given route.

We did two different experiments: the first one mined sequences in both peak hours simultaneously; the second one mined each peak hour considered individually (the morning and the evening ones). We did so to mine BBS peak-dependent (just occur in one of the two peaks), discovering whether the schedule points should be adjusted for the entire day or just in a specific period.

The results presented in Table 2 are for frequent subsequences of the HD sequences. We set PrefixSpan minimum support to 40% (sequences of length=1) and 20% (sequences with a length greater than 1) in the selected data partition, and a $ht=0.15$. We did so because the significance of the second case is higher than the first one. The second case demonstrates high correlations between distinct HD events in distinct bus stops that explain better the origin of the BB events.

5.1 Discussion

Firstly, we want to highlight that **only frequent HD subsequences (BBS) with events of type -1 (headway reductions) were detected**. All the sequences presents high confidence, demonstrating their specific validity in the bunching partition.

In route B1 two BBS were identified: BS2_B1 and the pair BS3_B1 and BS4_B1. Both are located at the beginning of the route: the gap verified in these points may become larger in successive stops. The pair is deeply analyzed in Table 3: the isolated

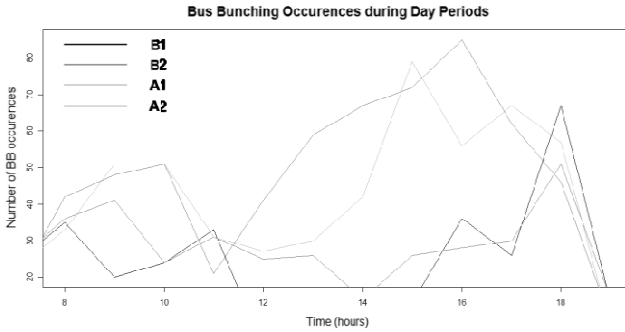


Fig. 6. Bus Bunching Occurrences during Day Periods. The trips with occurrences starting during the defined peak hours: 08:00-11:00 and 16:00-19:00.

Table 2. The values presented are the Support of the sequences (number of trips where those events occur / total number of BB trips considered) as well as the confidence between the occurrences of those in the trips with BB and the total trips occurred in the period

ID	Route	Peaks Considered	Sequence (possible BBS)	Support	Confidence
01	B1	Both	BS3_B1 = -1 BS4_B1=-1	0,2619	0,75
04	B1	Both	BS2_B1 = -1	0,4206	0,80
05	A1	Both	BS2_A1 = -1	0,5095	0,72
06	A2	Both	BS2_A2 = -1	0,5706	0,61
07	B1	8h to 11h	BS5_B1 = -1	0,4000	0,91
08	B1	8h to 11h	BS2_B1 = -1	0,4308	0,85
09	A1	8h to 11h	BS6_A1 = -1	0,4064	0,88
10	A1	8h to 11h	BS3_A1 = -1	0,4225	0,87
11	A1	8h to 11h	BS2_A1 = -1	0,5669	0,72
12	A2	8h to 11h	BS2_A2 = -1	0,6237	0,74
13	B1	16h to 19h	BS2_B1 = -1	0,4099	0,82
14	A1	16h to 19h	BS2_A1 = -1	0,4500	0,81
15	A2	16h to 19h	BS2_A2 = -1	0,6237	0,78

events in BS3_B1 and BS4_B1 have the same support than the events occurred in both bus stops. We can also set an association rule like BS3_B1= -1 -> BS4_B1= -1 (with a confidence of 97%) identifying a solid BBS in those two bus stops and an expected BB behavior. In Fig. 7, we illustrate one example of the pattern extracted on a morning peak hour of a typical working day. Assuming casual and regular passengers arriving, we describe two cases: (1 - Non-Bunching) ideal case: bus pair running with a short but regular headway; (2 - Bunching) real case: another pair running in the route with an irregular headway, having a BB event in BS10_B1.

Table 3. Detailed analysis of the mined sequence BS3_B1 = -1, BS4_B1=-1. The support of the highlighted sequences 01a and 01b are the same of the sequence 01: this can demonstrate an implication between the bus delays in the BS3_B1 and BS4_B1, an usual BB behavior. The confidence for a possible association rule BS3_B1 = -1 -> BS4_B1=-1 is 97%.

ID	Route	Peaks Considered	Sequence (possible BBS)	Support
01	B1	Both	BS3_B1 = -1 BS4_B1=-1	0,2619
01a	B1	Both	BS3_B1 = -1	0,2619
01b	B1	Both	BS4_B1 = -1	0,2619

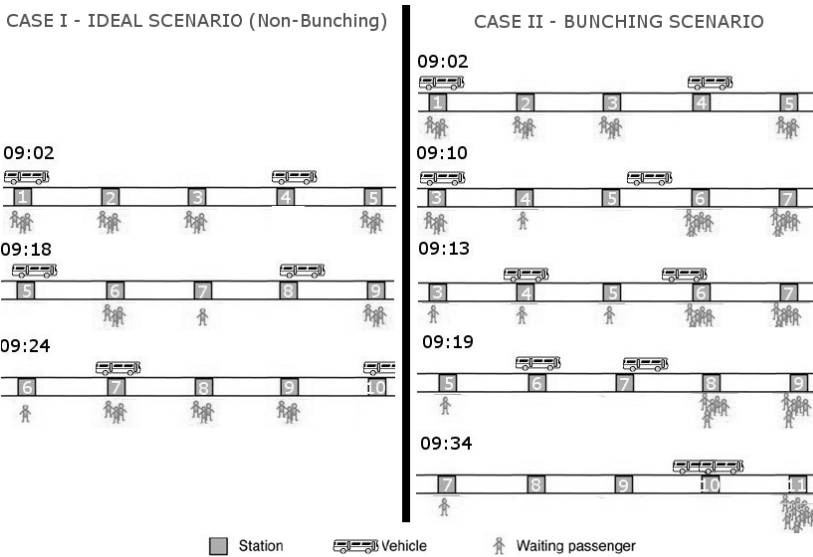


Fig. 7. Two possible cases in a Tuesday morning: one with BB and other without it. The numbers inside the squares are the bus stops’ identifiers. The case II is one of the 28,6% of BB trips with the frequent subsequence 01 (see Table 2). The passengers in each stop are an estimation assuming casual passenger arriving [8]. It is possible to observe the strong effect of the first HDs on the number of passengers waiting in the following bus stops and, consequently, in the headways.

In line A, BS2_A1 and BS2_A2 were identified as BBS. Additionally, they are - as well as the BBS identified in line B – located in the beginning of the route. The causes for this behavior are, probably, the large affluence of passengers in peak hours but the authors cannot sustain this with the available data.

Summarily, just BBS for the first bus stops were found. Based on this, we can conclude that **the BB in those routes were largely provoked by successive bus delays in the first bus stops** (the HD -1 events are mainly caused by bus delays [8]) although we cannot sustain whether they are failing the schedule.

In the second study, we analyzed whether the BBS identified were coherent in both peak hours. In route B1, the BS2_B1 is a BBS for both peak hours.

BS2_A1 and BS2_A2 are also persistent BBS in both peak hours. Those two bus stops correspond to an important bus interface (*Sá da Bandeira*) in the city and to a University Campus (*Asprela*), respectively. This happens because both routes maintain a high frequency and a large number of passengers during the day, being always busy.

In our opinion, the short lengths of the frequent subsequences mined (1 and 2) are not relevant compared with the relevance of the identified patterns. Those lengths will always depend on the routes analyzed, so they can be larger when applied to other datasets. The achieved patterns demonstrate that the BB patterns can be modeled like a frequent sequence mining problem. The results achieved demonstrate the utility of our framework to identify the exact schedule points to change in the timetables.

6 Conclusions and Future Work

In public transportation planning, it is crucial to maintain the passengers' satisfaction as high as possible. A good way to do so is to prevent the phenomenon known as Bus Bunching.

There are two main approaches to handle this problem: the PT planning one, anticipating and identifying the origin of the problem, and a real time one, which tries to reduce the problem online (during the network function).

Our approach is a contribution to solve the PT planning problem: this framework can help to identify patterns of bus events from historical data to discover the schedule points to be adjusted in the timetables.

In this paper, we presented a methodology to identify BB events that use headway deviations from AVL trips data. We ran a sequence mining algorithm, the PrefixSpan, to explore such data.

The results are promising. We clearly demonstrated the existence of relevant patterns in the HD events of the travels with bunching. There were some bus stops sequences along the routes identified as BBS - Bunching Black Spots, forming regions within the schedule points that should be adjusted. We want to highlight the following findings:

- The high correlation between HD in distinct bus stops – one event in a given bus stop provoke an event on another one with a regularity sustained by a reasonable support and confidence;
- The detection of BBS in the beginning of the routes demonstrated that HD that occurs in the beginning of the trips can have a higher impact into the occurrence of BB compared with events occurred in bus stops further.

The main contributions of this work are: 1) to model the BB trip usual pattern like a frequent sequence mining problem; 2) to provide the operator the possibility to mitigate the BB in a given line by adjusting the timetables, instead of suggesting forced actions that can decrease schedule reliability and, consequently, reduce passengers' satisfaction.

The identified patterns are no more than alerts that suggest a systematic cause for the BB in the studied routes. This information can be used to improve the schedule. The goal is not to eliminate those events but just to mitigate them. Our future work consists in forecasting BB in a data stream environment based on AVL data. By using this approach, the BSS will be identified online as the data arrive in a continuous manner [24]. This possibility will allow the use of control actions to avoid BB events that can occur even when the timetables are well adjusted, in order to prevent the majority of the potential BB occurrences.

Acknowledgements. We would like to thank STCP (Sociedade de Transportes Colectivos do Porto, S.A.) for the AVL historical data supplied to this work. We would also like to thank the support of the project Knowledge Discovery from Ubiquitous Data Streams (PTDC /EIA-EIA/098355/2008).

References

1. Gershenson, C., Pineda, L.: Why Does Public Transport Not Arrive on Time? The Pervasiveness of Equal Headway Instability. *PLoS ONE* 4 (2009)
2. Daganzo, C.: A Headway-Based approach to eliminate Bus Bunching. *Transportation Research Part B* 43, 913–921 (2009)
3. Pilachowski, J.: An approach to reducing bus bunching. PhD. Univ. of California, Berkeley, California (2009)
4. Lin, J., Ruan, M.: Probability-based bus headway regularity measure. *IET Intelligent Transport Systems* 3, 400–408 (2009)
5. Matias, L., Gama, J., Mendes-Moreira, J., Sousa, J.F.: Validation of both number and coverage of bus Schedules using AVL data. In: 13th International IEEE Annual Conference on Intelligent Transportation Systems, Funchal, Portugal, pp. 131–136 (2010)
6. Newman, P.: *Transit-Oriented Development: An Australian Overview*. Transit Oriented Development – Making it Happen (2005)
7. Strathman, J., Kimpel, T., Callas, S.: Headway Deviation Effects on Bus Passenger Loads: Analysis of Tri-Met’s Archived AVL-APC Data (2003)
8. Bellei, G., Gkoumas, K.: Transit vehicles’ headway distribution and service irregularity. *Public Transport* 2, 269–289 (2010)
9. Wang, F.: Toward Intelligent Transportation Systems for the 2008 Olympics. *IEEE Intelligent Systems* 18, 8–11 (2003)
10. Newell, G., Potts, R.: Maintaining a bus schedule. In: 2nd Australian Road Research Board, pp. 388–393 (Year)
11. Zhao, J., Dessouky, M., Bukkapatnam, S.: Optimal Slack Time for Schedule-Based Transit Operations. *Transportation Science* 40, 529–539 (2006)
12. Strathman, J., Kimpel, T., Dueker, K.: Automated bus dispatching, operations control and service reliability. *Transportation Research Record* 1666, 28–36 (1999)
13. Mishalani, R.: Passenger Wait Time Perceptions at Bus Stops: Empirical Results and Impact on Evaluating Real-Time Bus Arrival Information. *Journal of Public Transportation* 2 (2006)
14. Newell, G.: Control of pairing of vehicles on a public transportation route, two vehicles, one control point. *Transportation Science* 8, 248–264 (1974)

15. Powell, W., Sheffi, Y.: A Probabilistic Model of Bus Route Performance. *Transportation Science* 17, 376–404 (1983)
16. Nicholson, A., Mei, K.: Assessing the effect of congestion on bus service reliability. In: 2nd International Symposium on Transport Network Reliability, Christchurch, NZ (2004)
17. Furth, P., Hemily, B., Muller, T., Strathman, J.: Uses of Archived AVL-APC Data to Improve Transit Performance and Management: Review and Potential. *Transportation Research Board* (2003)
18. Vuchic, V.: *Transit Systems, Operations and Networks*. Urban Transit. Wiley, New York (2005)
19. Agrawal, R., Srikant, R.: Mining Sequential Patterns. In: Eleventh International Conference on Data Engineering, Taipei, Taiwan, pp. 3–14 (1995)
20. Srikant, R., Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. In: 5th International Conference on Extending Database Technology, Avignon, France, pp. 3–17 (1996)
21. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: 20th International Conference on Very Large Data Bases, Santiago de Chile, Chile, pp. 487–499 (1994)
22. Jian, P., Han, J., Mortazavi-asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. In: 17th International Conference on Data Engineering, Heidelberg, Germany, pp. 215–224 (2001)
23. TRB: Transit Capacity Quality of Service Manual. Transit Cooperative Research Program Web Document No. 6. *Transportation Research Board - National Research Council*, Washington, D.C. (1999)
24. Gama, J., Gaber, M.: *Learning from Data Streams*, New York (2007)

Detecting Abnormal Patterns in Call Graphs Based on the Aggregation of Relevant Vertex Measures

Ronnie Alves¹, Pedro Ferreira², Joel Ribeiro³, and Orlando Belo⁴

¹ Vale Technological Institute Sustainable Development, Brazil
ronnie.alves@vale.com

² Centre for Genomic Regulation, Spain
pedro.ferreira@crg.es

³ Eindhoven University of Technology, The Netherlands
j.t.s.ribeiro@tue.nl

⁴ University of Minho, Portugal
obelodi@di.uminho.pt

Abstract. Graphs are a very important abstraction to model complex structures and respective interactions, with a broad range of applications including web analysis, telecommunications, chemical informatics and bioinformatics. In this work we are interested in the application of graph mining to identify abnormal behavior patterns from telecom Call Detail Records (CDRs). Such behaviors could also be used to model essential business tasks in telecom, for example churning, fraud, or marketing strategies, where the number of customers is typically quite large. Therefore, it is important to rank the most interesting patterns for further analysis. We propose a vertex relevant ranking score as a unified measure for focusing the search of abnormal patterns in weighted call graphs based on CDRs. Classical graph-vertex measures usually expose a quantitative perspective of vertices in telecom call graphs. We aggregate wellknown vertex measures for handling attribute-based information usually provided by CDRs. Experimental evaluation carried out with real data streams, from a local mobile telecom company, showed us the feasibility of the proposed strategy.

1 Introduction

Graphs have become increasingly important in modeling sophisticated structures and their interactions in a large variety of applications, ranging from chemical informatics to telecommunications [3]. Particularly in the latter, business analysts can make use of graph-based analysis for better understanding customer social behavior and thus devising proper business strategies. For instance, from a business point of view, it has been shown that it is more reasonable to retain (or maintain) existing customers rather than acquiring new ones [5]. If the company anticipates the intention of the customer to leave (typically called “churn”), proper measures can be taken to avoid such action. In the telecom

context, customers can be seen as vertices (nodes) of the network graph and the calls made between them the edges (arcs). An edge connecting two customers contains the information that for a given instant summarizes the calling pattern between them. This data structure is called a call graph being particularly large and sparse [7].

Let's assume a typical scenario where the business analyst wants to search for potential fraud situations, i.e., looking for customers presenting abnormal pattern behaviors. Chances are that their vertices act quite distinctively in the entire graph, and thus, relevant vertices, which could be acting as fraudsters should present a particular behavior. Indeed, a relevance vertex measure needs to take into consideration attribute-based and structural vertex measures. Classical graph-vertex measures usually expose quantitative information of the vertices in the graph. From the extensive literature in mining call graph patterns we highlight two works, which employ different graph measures for better understanding of call graphs [4,7]. More recently, in [2] was presented a complete survey about graph mining, covering algorithms, laws and generators. There are also other kinds of graph patterns based on frequent patterns analysis [9]. With respect to graph mining over CDRs, Cortes et al. [4,3,9] propose a data structure based on the union of small sub graphs (Top-K edges), called community of interests (COI) to handle large dynamics graphs. Different from all those previous studies we make a clear distinction between attribute-based and "purely" structural graph-vertex measures.

In this work we do not intend to compare the effectiveness of several graph-vertex measures for scoring interesting (abnormal) vertices. Rather, we propose a unified ranking strategy which makes use of enhanced classical vertex measures combining attribute-based information and graph structural information, aggregating vertex measures into a unified measure for revealing abnormal patterns in call graphs. This network was also explored in other studies on telecom fraud detection by exploring customer behavior using signatures [6] and dynamic clustering [1]. The main contributions of the proposed strategy are: 1) a dynamic model for mining evolving call graph networks, so the model can be up-to-date when new information is available; b) a set of relevant vertex measures devised for allowing attribute-based and structural evaluation of call graphs; and 3) a vertex ranking function for mining abnormal K-vertices by applying a unified strategy that aggregates distinct vertex measures of interestingness. The remainder of this paper is organized as follows. In Section 2 we present the concepts to define the structure and properties of call graphs. In Section 3 the interest measures used in this work are described. Next, in Section 4 the proposed mining strategy is explained followed by an empirical evaluation in Section 5. Conclusions and future work are provided in Section 6.

2 Evolving Call Graphs

Before presenting the proposed mining strategy we need first to highlight a few concepts to understand the problem of evaluating abnormal patterns by

aggregating attribute-based and structural graph vertex-measures in evolving call graphs.

Annotated Call Graph. An annotated call graph is a digraph $G' = (V, E, A)$, where

- V is a set of vertices or nodes,
- E is a set of ordered pairs of vertices, called directed edges,
- A is a powerset of attribute-based information that describes the edges.

Each edge $e(x, y, info) \in E$ (with $info \in A$) denotes a direct connection from x to y and contains a set of attribute values ($info$).

Due to the dynamic nature of telecom networks, data is being obtained constantly from new calls. In order to capture and reflect the new data the concept of evolving call graph is introduced [3].

Evolving Call Graphs. An evolving call graph consists of an annotated call graph updated with new information. The weight of the new and the old information is defined by a weighting factor θ . For a given instant t , the new call graph G'_t reflects the information and the structure of the graph in the previous instant G'_{t-1} and the new information g'_t , as described in Eq. 1.

$$G'_t = \theta \cdot G'_{t-1} \oplus (1 - \theta) \cdot g'_t \quad (1)$$

The weighting factor θ models the longevity of the information, i.e., how long the information that represents a call is reflected in the graph. Since the model can be updated over time with new CDRs, we are able now to formulate the process of mining abnormal patterns (vertices or customers) in call graphs by aggregating attribute-based and structural graph vertex measures.

Problem: Mining Abnormal Usage Patterns in CDRs.

Given the information provided by a transaction database of CDRs T and the respective evolving graph G' , find customers (vertices) that present an abnormal pattern. This should take into account the attribute and structure information from G' . Such aggregation model should assist the evaluation of vertex relevance in G' according to a unified vertex ranking measure.

3 Relevance Vertex Measures

A vertex is said to be relevant on a graph when its behavior distinguishes from the other vertices, either from a structural or attribute-based perspective on the entire graph. Attribute-based measures evaluate a vertex according to the information associated to its incident edges, for example call duration and billed time. Structural measures evaluate a vertex with respect to its interaction(s). A well-known structural measure used to evaluate the relevance of vertices (web

pages) in a web graph is the PageRank [8], used for instance in the Google ranking scheme. In the telecom context, vertices with high page rank may reveal clients who have a high social importance and due to their economical importance should be the target of particular inspection [7].

In this work we distinguish relevance vertex measures as purely structural and attribute-based ones. Purely structural are vertex measures that take into account the graph structure (e.g., degree of centrality). Attribute-based measures can be also context-sensitive (e.g., closeness centrality) or context insensitive (e.g., vertex-usage being explained in the next section).

3.1 Vertex Usage (M_1)

The Vertex Usage measure is based on the standard score (or z-score) used in statistics, and indicates how much the behavior of a graph vertex (observation) deviates from the mean of the entire graph. A vertex presenting a high vertex usage score may be viewed as abnormal pattern, being potential indicative of fraud or churning situations in telecom. Vertex Usage of a vertex v and an attribute i is defined as follows:

$$M_1(v, i) = \frac{x_i - \bar{x}_i}{\sigma_i} \tag{2}$$

where $i \in info$, x_i is the observation of i in v , \bar{x}_i is the mean of i , and σ_i is the standard deviation of i . The computational cost to calculate this measure is $O(I \cdot V)$, being I the set of attributes and V the set of vertices.

As an example of its application let's use the information of Table 1 and assume that we want to evaluate the Vertex Usage for the entire vertices of the graph G'_t . This table presents the statistics associated to the attributes *Air Time* (AT) and *Charged Amount* (CA) used for the Vertex Usage score. Vertex Usage results of each attribute, i.e., the values of each attribute for incoming, outgoing and all edges of all vertices of the graph, are shown in Table 2.

Table 1. Example of statistics of two attributes of a call graph G'_t

Edge		Attribute-based <i>info</i>	
Origin	Dest.	AT	CA
1	2	51.00	13.60
2	3	285.00	152.05
2	4	8.25	2.40
AVG		114.75	56.02
STD		121.64	68.06

3.2 Degree of Centrality (M_2)

The degree of centrality is a structural measure defined as the ratio of the *incoming* and *outgoing* edges of a vertex, and the total number of edges on the

Table 2. Vertex usage (M_1) values of the vertices in Table II

v	$M_1(v, AT)$			$M_1(v, CA)$		
	Ori.	Des.	All	Ori.	Dest.	All
1	-0.52	-0.94	-0.52	-0.62	-0.82	-0.62
2	1.47	-0.52	1.89	1.45	-0.62	1.65
3	-0.94	1.40	1.40	-0.82	1.41	1.41
4	-0.94	-0.88	-0.88	-0.82	-0.79	-0.79

graph. Vertices with high scores may be viewed as cases of popularity where a relationship exists with many other vertices. These relationships should not be seen as a unique factor of vertex importance since this measure only considers the links without their attribute-based information. The degree centrality of a vertex v is defined as follows:

$$M_2(v) = \frac{inDegree(v) + outDegree(v)}{count(edges)} \quad (3)$$

where $inDegree(v)$ and $outDegree(v)$ are the count of incoming and outgoing incident edges of v , and $count(edges)$ is the total number of edges in the graph. Table III depicts the computational process of this measure.

Table 3. Degree of centrality (M_2) values of the vertices in Table II

	Vertex v			
	1	2	3	4
$inDegree(v)$	0	1	1	1
$outDegree(v)$	1	2	0	0
$M_2(v)$	0.33	1.00	0.33	0.33

3.3 Closeness Centrality (M_3)

Closeness centrality can be considered either as attribute-based (when using information associated to its edges) or structural-based (when counting only the existence of edges) graph vertex measure. Closeness centrality indicates how close a vertex is on average to all other vertices and it is defined as follows:

$$M_3(v, i) = \frac{\sum_{k=1}^n \min Dist(v, reachableVertex(v, k), i)}{reachableVertices(v)} \quad (4)$$

where $i \in info$, $reachableVertices(v)$ is the count of reachable vertices of v , $reachableVertex(v, k)$ is the k^{th} reachable vertex of v , and $\min Dist(v_1, v_2, i)$ is the minimum distance between the vertices v_1 and v_2 with respect to i . Table IV illustrates the calculation of this measure. For this measure we assume

the set of attributes $I = \{Occurrences, Air\ Time, Charged\ Amount\}$ where $Occurrences(OC)$ is the number of edges (when the number of reachable vertices is zero we directly assign this value as M_3 result).

Table 4. Closeness Centrality (M_3) values of the vertices in Table 1

	Vertex v			
	1	2	3	4
$reachableVertices(v)$	3	2	0	0
$M_3(v, OC)$	1.67	1.00	0.00	0.00
$M_3(v, AT)$	148.75	146.63	0.00	0.00
$M_3(v, CA)$	65.08	77.23	0.00	0.00

3.4 Vertex Interest (M_4)

This attribute-based measure is inspired on the PageRank measure. In telecom context, a vertex will have a high rank if receives many calls or alternatively if these calls have a high importance to the origin vertex. The vertex interest $M_4(v, i)$ of a vertex v and an attribute i is defined as follows:

$$\alpha \cdot \sum_{a=1}^k \frac{inEdgeValue(v, a, i)}{vertexAVG(v, i)} + \beta \cdot \sum_{b=1}^k \frac{outEdgeValue(v, b, i)}{vertexAVG(v, i)} \tag{5}$$

where $i \in info$, α and β are user-defined constants, $vertexAVG(v, i)$ is the average of the attribute i for all incident edges of v , $inEdgeValue(v, k, i)$ is the value of the k^{th} incoming incident edge of v , $outEdgeValue(v, k, i)$ is the value of k^{th} outgoing incident edge of v , a is the total number of the incoming incident edges of v , and b is the total number of the outgoing incident edges of v . An example of its application is given in Table 5.

Table 5. Vertex Interest (M_4) values of the vertices in Table 1

	Vertex v			
	1	2	3	4
$vertexMean(v, AT)$	51.00	114.75	285.00	8.25
$vertexMean(v, CA)$	13.60	56.02	152.05	2.40
$M_4(v, AT)$	1.00	3.00	1.00	1.00
$M_4(v, CA)$	1.00	3.00	1.00	1.00

4 Finding Abnormal Vertices

Since $info$ may contain several attributes, the computation of a unified relevance vertex measure can result in a set of different measures. Thus, the aggregation

function I_{AGG} to combine (aggregate) all relevance vertex attributes (information) is defined by:

$$I_{AGG}(v, I, m) = \max [norm(M_m(v, i))] \text{ for all } i \in I \quad (6)$$

where I is a set of attribute values, $m \in \{1, 2, 3, 4\}$ according to a relevance vertex measure, and $M_m(v, i)$ is the value of a measure of interest m of the vertex v . $norm(x) \in [0, 1]$ refers to the min-max normalization of x . Table 6 illustrates the I_{AGG} calculation.

Table 6. Aggregation of the attributes *Air Time* and *Charged Amount* with the measure M_3 , according to Table 4

	Vertex v			
	1	2	3	4
$M_3(v, AT)$	148.75	146.63	0.00	0.00
$M_3(v, CA)$	65.08	77.23	0.00	0.00
$norm(M_3(v, AT))$	1.000	0.986	0.000	0.000
$norm(M_3(v, CA))$	0.843	1.000	0.000	0.000
$I_{AGG}(v, \{AT, CA\}, 3)$	1.000	1.000	0.000	0.000

Considering all vertices v , $I_{AGG}^k(I, m)$ refers to the top- k $I_{AGG}(v, I, m)$ values. Taking into account only the Top-K cases, it is possible to identify the most interesting vertices through Eq. 7.

$$I_{AGG}^k(v, I, m) = \begin{cases} I_{AGG}(v, I, m) & \text{if } I_{AGG}^k(I, m) \text{ contains } v \\ \min I_{AGG}^k(I, m) & \text{otherwise} \end{cases} \quad (7)$$

Telecom call graphs are usually big figures for being explored at one shot. Therefore, one should be able to focus on particular spots of the entire graph. To do so, the graph composition function M_{AGG} for measure aggregation according the Top-K vertices is defined (Eq. 8). Such function compares a specific vertex with the Top-K vertices for each relevance vertex measure using all vertex measures at once, being possible to constraint the search of (Top-K) abnormal patterns in the entire graph.

$$M_{AGG}(v, I, k) = \prod_{m=1}^n I_{AGG}^k(v, I, m) \quad (8)$$

Remark that only Top-K values of each measure are used. An example of M_{AGG} calculation is presented in Table 7 (it is assumed that $k = 2$), where vertices 2 and 3 are the most interesting ones.

The first step to calculate the M_{AGG} score is evaluating I_{AGG}^2 values for each measure $m = \{1, 2, 3\}$. For Vertex Usage ($m = 1$) we refer to Table 2 to observe all scores (see column ‘‘All’’) for all vertices, where I_{AGG}^2 is evaluated as follows:

- $I_{AGG}^2(v = 1, \{Air\ Time, Charged\ Amount\}, 1)$
 $= \max(0.130, 0.070) = 0.130$
- $I_{AGG}^2(v = 2, \{Air\ Time, Charged\ Amount\}, 1)$
 $= \max(1.000, 1.000) = 1.000$
- $I_{AGG}^2(v = 3, \{Air\ Time, Charged\ Amount\}, 1)$
 $= \max(0.823, 0.902) = 0.902$
- $I_{AGG}^2(v = 4, \{Air\ Time, Charged\ Amount\}, 1)$
 $= \max(0.000, 0.000) = 0.000$

Finally the two highest scores for Vertex Usage are $I_{AGG}^2 = \{1.000, 0.902\}$. The I_{AGG}^2 scores for Closeness Centrality ($m = 3$; Table 4) are $I_{AGG}^2 = \{1.000, 0.986\}$. The Vertex Interest scores ($m = 4$; Table 5) are $I_{AGG}^2 = \{1.000, 0.333\}$. The M_{AGG} for $v = 1$ is then evaluated as:

$$\begin{aligned}
 & - M_{AGG}(v = 1, \{Air\ Time, Charged\ Amount\}, 2) \\
 & = 0.902 \times 1.000 \times 0.333 = \\
 & = 0.300
 \end{aligned}$$

Table 7. Measure (M_{AGG}) and information (I_{AGG}) aggregation for the measures {Vertex Usage, Closeness Centrality, Vertex Interest} and the attributes {*Air Time*, *Charged Amount*}, according to Table 1

	Vertex v			
	1	2	3	4
$I_{AGG}^2(v, \{AT, CA\}, 1)$	0.130	1.000	0.902	0.000
$I_{AGG}^2(v, \{AT, CA\}, 3)$	1.000	0.986	0.000	0.000
$I_{AGG}^2(v, \{AT, CA\}, 4)$	0.333	1.000	0.333	0.333
$M_{AGG}(v, \{AT, CA\}, 2)$	0.300	0.986	0.296	0.296

Table 8. Statistics about all call graphs in the related week sample. Each sample corresponds to a particular day.

Sample	Vert.	Edges	Comp.	Diam.	Path	AvgNeig.
1	35726	20356	15564	2	1.014	1.131
2	22886	12270	10688	3	1.010	1.067
3	22377	11896	10531	3	1.007	1.059
4	21743	11508	10287	3	1.005	1.054
5	21956	11598	10426	2	1.008	1.051
6	22100	12016	10142	2	1.005	1.083
7	20380	11234	9237	2	1.004	1.097

5 Detecting Potential Fraud Situations in Call Graphs

In this section we present a case studied using real data streams from a mobile telecom company. The main goal was to highlight potential fraud situation using

the proposed strategy. It was provided a list of fourteen fraud cases obtained from a specific week of CDRs. For each day of the week there are approximately 2.5 millions of records (CDRs) and 700,000 customers. In this empirical study only around 5% of the entire dataset containing both fraud (fourteen situations) and potential unidentified cases were selected for further analysis. Table 8 provides statistics about the call graphs obtained for each day of the week. As one can observe these graphs are quite sparse, being a great challenge the detection of abnormal call patterns.

Table 9. Results of the vertex ranking on the final evolving call graph for the known cases

	Blacklist Cases													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
M_1						16		6	62	33	84	82	5	10
M_2		67	54	31		24		38	50		31	21		67
M_3														
M_4		71	46	34		23		36	48		39	21		73
M_{AGG}			49	51		16		17	30		26	23	50	42

Further discussion about variables related to this dataset can be found in previous works [16]. In order to assess effectiveness, all relevance vertex measures are computed for all vertices of the 5% sample. Then, it was verified whether fraud cases are in the Top-100 results or not. The Degree of Centrality is able to highlight eleven cases of fraud maybe due to the increasing number of calls in that week. On the other hand the Closeness Centrality measure does not detect any case of fraud consequence to the lower diameter and average neighborhood of the call graphs (Table 8).

Table 10. Results of the vertex ranking on the daily call graphs for the known cases

	Blacklist Cases													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
M_1					69	6		8	7	11	40	40	2	1
M_2	27	11	15	9		5		11	15	29	14	9	37	20
M_3					52		45	74		5				6
M_4	26	13	14	12		4		10	16	42	12	9	31	23
M_{AGG}	41	19	19	12		3		5	6	22	8	9	11	5

Tables 9 and 10 present the results of applying all relevance measures employed in the proposed aggregation strategy. For each of the fraud cases, it is identified their ranking in the Top-100 of the different measures. Table 9 refers the application of vertex ranking taking into account the preference selection function (Eq. 8) on the final evolving call graph (i.e., aggregating all samples).

Table 11. Results of the detection of abnormal patterns using different graph-based metrics. I = Information, S = Structure, B = Both

	Blacklist Cases													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Evolving call graph		B	B	B		B		B	B	I	B	B	I	B
Daily call graphs	B	B	B	B	S	B	S	B	B	B	B	B	B	B

Similarly, Table 10 refers the application of vertex ranking but for each of the samples (i.e., each sample as a separate call graph). Each value represents, for a given case and measure, the highest ranking for the different samples.

The results suggest that there are different types of fraud situations. One of our conclusions is that some cases can be grouped according their rankings. Probably different groups identify distinct types of fraud and should be differently handled by the fraud analysts. One strategy should be the selection of other similar sub-graphs based on such groups identified by the proposed model 10 11. Indeed, for detecting abnormal patterns in weighted call graphs one should not set aside attribute-based information about the calls. This observation explains why Vertex Usage and Vertex Interest are more sensitive to this type of problem.

The application of the vertex ranking on the final evolving call graph (Table 9) identified successfully 9 out of 14 (around 65%) of the given cases as high potential fraud cases. The same application on the daily call graphs (Table 10) improved the detection rate to 86%. Finally in Table 11 we summarized the fraud detection analysis. The conclusion is that the evolving call graph (G'_t) should be taking into account together with the daily call graphs (g'_t).

6 Conclusions

In this work we have presented enhancements on well-known graph-vertex measures in order to improve selection and ranking of abnormal patterns over telecom call graphs. We extend classical quantitative vertex measures with attributed-based ones, proposing a unified vertex ranking for detecting abnormal vertices in weighted graphs. An empirical study using CDRs from a real mobile telecom company showed us the feasibility of the proposed strategy, while recovering most of the potential fraud situations.

References

1. Alves, R., Ferreira, P.G., Belo, O., Lopes, J., Ribeiro, J., Cortesao, L., Martins, F.: Discovering telecom fraud situations through mining anomalous behavior patterns. In: Proceedings of the DMBA Workshop, on the 12th ACM SIGKDD (2006)
2. Chakrabarti, D., Faloutsos, C.: Graph mining: Laws, generators, and algorithms. ACM Comput. Surv. 38(1), article 2 (June 2006)

3. Cortes, C., Pregibon, D., Volinsky, C.: Communities of interest. *Intelligent Data Analysis* 6, 211–219 (2002)
4. Cortes, C., Pregibon, D., Volinsky, C.: Computational methods for dynamic graphs. *Journal of Computational and Graphical Statistics* 12, 950–970 (2003)
5. Euler, T.: Churn Prediction in Telecommunications Using Mining Mart. In: *Proceedings of the DMBiz Workshop, on the 9th European Conference on Principles and Practice in Knowledge Discovery in Databases, PKDD 2005* (2005)
6. Ferreira, P., Alves, R., Belo, O., Cortesão, L.: Establishing Fraud Detection Patterns Based on Signatures. In: Perner, P. (ed.) *ICDM 2006*. LNCS (LNAI), vol. 4065, pp. 526–538. Springer, Heidelberg (2006)
7. Nanavati, A.A., Gurumurthy, S., Das, G., Chakraborty, D., Dasgupta, K., Mukherjee, S., Joshi, A.: On the structural properties of massive telecom call graphs: findings and implications. In: *Proceedings of CIKM 2006*, pp. 435–444 (2006)
8. Page, L., Brin, S., Motwani, R., Winograd, T.: *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report. Stanford InfoLab. (1999)
9. Pennock, D.M., Flake, G.W., Lawrence, S., Glover, E.J., Giles, C.L.: Winners don't take all: Characterizing the competition for links on the web. In: *Proceedings of Proc. Natl. Acad. Sci. USA*, pp. 5207–5211 (2002)
10. Akoglu, L., McGlohon, M., Faloutsos, C.: *oddball*: Spotting Anomalies in Weighted Graphs. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) *PAKDD 2010*. LNCS, vol. 6119, pp. 410–421. Springer, Heidelberg (2010)
11. Aggarwal, C.C., Zhao, Y., Yu, P.S.: Outlier detection in graph streams. In: *Proceedings of ICDE 2011*, pp. 399–409 (2011)

Real-Time Mass Flow Estimation in Circulating Fluidized Bed

Andriy Ivannikov¹, Mikko Jegeroff², and Tommi Kärkkäinen¹

¹ Department of Mathematical Information Technology, University of Jyväskylä,
P.O. Box 35, FIN-40014, Jyväskylä, Finland
andriy.v.ivannikov@student.jyu.fi, tommi.karkkainen@jyu.fi

² VTT, Technical Research Centre of Finland,
P.O. Box 1603, FIN-40101, Jyväskylä, Finland
mikko.jegeroff@vtt.fi

Abstract. The mass flow parameter identification is important for modeling and control purposes in Circulating Fluidized Bed technology. In this article we propose a novel method for estimating the mass flow in the Circulating Fluidized Bed and consider aspects of its application. The method is based on combining information obtained from both mass of fuel silo and velocity of fuel screw signals. The information from mass of fuel silo measurements is extracted by following the lower edge of the signal.

Keywords: CFB, control, estimation, mass flow, system identification.

1 Introduction

The mass flow (g/s) parameter plays an important role in modeling and control of Circulating Fluidized Bed (CFB). As the direct measurement of mass flow is not usually available, its estimation reveals an important and challenging problem. Online aspect of the developed techniques receives special emphasis as the estimates of mass flow should be available in real-time mode for control purposes.

The major elements of the fuel feeding system are (1) the fuel tank, (2) the feeding screw, and (3) the mixing screw. The scales are located underneath the tank. The readings of the scales become immediately available in the control system in electronic form as well as the velocities (rpm) of the feeding and mixing screws.

There are two main operation periods: (1) fuel feeding, (2) fuel consumption (see Fig. 5(a)). During the fuel feeding period a new portion of fuel is fed to the tank, while consumption also continues. This results in a sharp increase of the fuel mass signal. During the fuel consumption phase fuel is only consumed or fed to the boiler. This corresponds to a slope line periods. The control system notifies explicitly when the feeding starts and ends by setting a binary flag CLOSED to either zero or one.

The process of fuel consumption is not just as simple as piece-wise linear, however. The rate of fuel feeding depends on the amount of the fuel in the fuel tank. The more fuel is in the tank the larger is the pressure on the lower layers of fuel mass and,

hence, the higher is the rate of fuel feeding for the same velocity of the feeding screw. Therefore, even under assumption of fuel homogeneity the mass flow process has nonzero second order derivatives.

Besides the pressure-related second order phenomena there are also many other process events and conditions influencing the immediate rate of the fuel consumption: (1) the change of feeding screw speed that visually results in the change of the slope of the mass measurement, (2) heterogeneity of the fuel (different sizes of the fuel particles, inhomogenous mixtures of different fuels), (3) sharp change of fuel (one fuel is in the bottom and another one is on top, i.e. no mixing) that results in an effect similar to the feeding screw rpm change. All these factors present major challenges in control of CFB as they invoke transient processes which are harder to control.

The fuel feeding system is non-rigid and subject to shakes. Scales do not compensate for possible movements of the fuel subsystem parts. As the result, the measured mass of fuel silo signal is contributed by several forces. Besides the mass proper, in the measured fuel mass signal there are at least (1) oscillations that are caused by the vibrations of the fuel subsystem parts originated from the rotation of the feeding and mixing screws and (2) peaks going strictly upwards that appear due to the fuel particle jamming in the feeding screw that causes short-term shocks to the fuel feeding subsystem (see Fig. 1 and Fig. 5(a)). The oscillations are of the same frequencies as the rotation frequencies of the feeding and mixing screws.

If the signal were free of vibration and jamming artifacts, then the instantaneous mass flow could simply be approximated by differentiating the fuel mass measurement. However, as the true mass signal is mixed up with the noise one needs to separate mass-related component from the noise prior to differentiation.

Several attempts have been made to address problem of online mass flow estimation [1]-[3]. However, despite the sophisticated and advanced approaches used in these methods they were not sufficiently accurate and fast enough. The methods were concentrated around the upward artifact detection strategy that primarily defined performance properties. For example, non-parametric methods for hunting peak artifacts are rather slow, likewise parametric methods are less reliable. In our approach we explicitly avoid dealing with upward peak artifacts by following only the lower edge of the fuel mass measurement for the extraction of mass-related component of the signal and estimation of fuel-specific aspect of mass flow, i.e. related to heterogeneity of fuel silo. In addition, this information is combined with the rpm of feeding screw readings to enable faster response to the changes in fuel feeding rate not dependent on the fuel homogeneity properties.

2 Method Description

We distinguish global process time and the local process time. As the name suggests, global process time is the time passed since the launch of the operation. Likewise the local time is the time passed since the start of the last fuel consumption period, i.e., the local time is reset to zero each time a new consumption phase begins. In our computations we always consider only local time denoted by t .

Assume that the mass of fuel silo measurement is denoted by $m(t)$, where t denotes time. The measurements of the feeding screw velocity are denoted as $\text{rpm}(t)$, which we call instantaneous or immediate velocity. However, in order to smooth rpm signal against noise effects in our computations we often use the estimate of the feeding screw velocity denoted by $\widehat{\text{rpm}}(t)$ and computed as

$$\widehat{\text{rpm}}(t) = \frac{1}{N_{\text{rpm}}} \sum_{\tau=t-N_{\text{rpm}}+1}^t \text{rpm}(\tau).$$

Therefore, at each sample of time (unless the immediate rpm equals zero) the current rpm value $\widehat{\text{rpm}}(t)$ is estimated as the average of the immediate rpm values from the last N_{rpm} time samples. For this purpose a queue (of maximal size $N_{\text{rpm}}^{\text{max}}$) of last N_{rpm} immediate rpm values is stored in memory and constantly updated during the operation.

If the current immediate rpm falls to zero, then the estimated $\widehat{\text{rpm}}(t)$ is also hard set to zero by zeroing the queue of past rpm values.

Due to the noise effects the measured immediate rpm can be negative or show small positive value even, if the real rpm is zero. Due to this fact we hard threshold the measured values of rpm to be zero any time, when the measured rpm value is less or equal to 0.062 rpm that was ascertained empirically.

The core of the method is based on tracking the lower edge of the mass of fuel silo measurement. This allows approximating the true mass signal while disregarding the artifacts, which occur only upwards. Therefore there is no need for peak artifact detection strategy. Previously peak detection was addressed by non-parametric or parametric approaches that were either slow or insufficiently accurate.

For this purpose we introduce a concept of the current lowest sampled point, i.e., the point, where the lowest value of mass of fuel silo measurement was observed since the beginning of the last consumption period. We denote the time, when a new lowest point was observed by $t_l(t)$, which is a function of time itself (we will skip explicit dependence on t , if $t_l(t) = t$ is assumed).

When a new consumption period starts, the current lowest point is initialized by the first value of $m(t)$ in the period, i.e., $m(1)$. The current lowest point is always stored in memory. Each time a new lowest point arrives, it replaces the old one in the memory. Moreover, if the difference $t_l - t_l(t-1)$ is greater than or equal to a predefined threshold value Δt_{min} ($t_l - t_l(t-1) \geq \Delta t_{\text{min}}$) then the local estimate of the mass flow is computed as

$$k(t) = \frac{m(t_l) - m(t_l(t-1))}{t_l - t_l(t-1)} \frac{1}{T_s}.$$

Geometrically interpreting, a hypothetical line is drawn between the previous and the current lowest points, and a local mass flow is computed as the slope of this line multiplied by the sampling rate $f_s = \frac{1}{T_s}$ (Hz), where T_s (sec) denotes the sampling time interval (see Fig. 1). This estimate is computed over a relatively short time interval and is generally considered as not reliable, hence, the name local. The newly

computed local g/s estimate is recorded to the queue of maximal size N_k^{max} containing past N_k local estimates of g/s. According to the queue definition, if the queue is full already, the oldest value is pushed out by the newest one.

If the condition $t_i - t_l(t - 1) \geq \Delta t_{min}$ is not satisfied, the algorithm performs the same as if no new local g/s was computed.

The use of Δt_{min} parameter is necessary to ensure that only statistically reliable and plausible local g/s estimates are used in global g/s computation. This becomes especially important when the queue of local g/s is small yet and global g/s estimate is too much sensitive to each individual local g/s estimate stored in the queue. The natural measure of reliability is the time interval during which the local mass flow estimate was computed, hence, the threshold is imposed on it.

The time, when the last new local mass flow estimate was computed and the queue of the latest g/s estimates was updated, is denoted by $t_u(t)$, which is a function of time itself (we will further skip explicit dependence on t , when $t_u(t) = t$ is assumed).

The estimate of the velocity of the feeding screw at this time is memorized and is called the last/previous informative rpm and is denoted by $\widehat{rpm}(t_u(t))$. It is initialized by zero value.

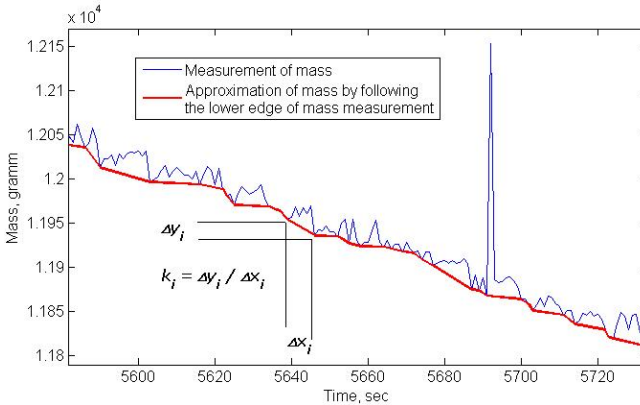


Fig. 1. Example of local mass flow estimation. Illustration of noise types: oscillations and the peak artifact.

During the consumption period the current global estimate of g/s is computed as the weighted sum of the local mass flow estimates stored in the queue:

$$\hat{k}(t) = \left| \frac{\sum_{i=1}^{N_k} k_i w_i}{\sum_{i=1}^{N_k} w_i} \right|,$$

where the weights w_i are the time intervals during which the corresponding local mass flows stored in the queue k_i were estimated, i.e., these are the projections of the hypothetical lines, whose slopes are k_i , onto the time axis. Formally weights are expressed as $w_i = t_i - t_l(t_i - 1)$, where t_i is the time when the respective k_i was

computed. Weights are also stored in the memory in the same type of queue as the local mass flow estimates. If the queues of k_i and w_i are not full yet then the sums are computed over smaller than N_k^{max} set of size N_k of existing elements. This notion concerns all arrays used in the algorithm, for example, also the queue of immediate rpm values.

The weights here perform the role of weighting the local g/s estimates according to their reliability that is measured by w_i .

Preliminarily, before the $\hat{k}(t)$ computation, if $\widehat{rpm}(t_u(t-1))$ is not equal to zero, then the queue of local mass flow estimates is updated by multiplying each k_i in the queue by the fraction of change in rpm since the last informative rpm was computed:

$$k_i = k_i \frac{\widehat{rpm}(t_u)}{\widehat{rpm}(t_u(t-1))}, \forall i = 1, \dots, N_k.$$

The latter operation is done in order to increase the sensitivity of global g/s estimate to the immediate changes in the fuel consumption process reflected by the rpm readings.

When the measured immediate rpm of the feeding screw falls to zero, the algorithm assumes that the process of fuel consumption is interrupted, stops all computations and starts outputting zero mass flow estimate $\hat{k}(t) = 0$. Moreover, all queues and counters accumulating history of the previous operation are reset to their initial zero values, except \hat{k}_{rpm} (see definition later on) and the local time t timer. When the rpm begins to rise again, then the computational process starts from the beginning. Moreover, local time timer is reset to zero, if the rotation of the fuel feeding screw is resumed at the consumption period.

There is a fuel-specific quantity k_{rpm} that expresses the mass flow per 1 rpm. Thus, it has units $\frac{g/s}{rpm}$. In case if the queue of local mass flow estimates is empty, this quantity can help to obtain estimates of global mass flow, which are close to real values. For example, when the control process is just started or continued after a break there are no entries in the queue of local mass flow estimates available.

The estimate of k_{rpm} is constantly updated and the last value of it is stored in memory during the operation. Namely, a new value of k_{rpm} estimate \hat{k}_{rpm} , which is considered most reliable for a current time, is computed once a new local mass flow estimate is obtained, i.e., new lowest point with weight $w_i \geq \Delta t_{min}$ arrives. \hat{k}_{rpm} is initialized by zero in the beginning of the control process and is never reset even if the immediate rpm falls to zero.

When the conditions for \hat{k}_{rpm} update are fulfilled it is computed as

$$\hat{k}_{rpm} = \hat{k}(t_u) / \widehat{rpm}(t_u),$$

i.e., a new current global mass flow estimate divided by the current rpm estimate.

Moreover, the estimated k_{rpm} is updated only if the same or more reliable estimate can be computed. The natural parameter that expresses the reliability of current k_{rpm} estimate is the number K_{used} of local mass flow estimates that were used for

computing the current/last k_{rpm} estimate. Thus, update of \hat{k}_{rpm} is only conducted, when the number $K_{current}$ of local g/s estimates residing in the queue is greater or equal to the K_{used} .

At every time sample, if at least one of the following conditions is satisfied: (1) the consumption period just started and this is not the start of the process of computations, (2) there is no new lowest point during the consumption phase, (3) feeding stage, (4) a new lowest point $m(t_i)$ was observed, but its weight $w < \Delta t_{min}$ - the computations are done as follows. If $r\widehat{pm}(t_u(t))$ is not zero, i.e., queue of local mass flow estimates is not empty then

$$\hat{k}(t) = \left| \frac{r\widehat{pm}(t)}{r\widehat{pm}(t_u(t))} \hat{k}(t_u(t)) \right|.$$

Otherwise, if the previous informative rpm $r\widehat{pm}(t_u(t))$ equals zero or the process just started, then the global mass flow is estimated as the product of the estimate of k_{rpm} and the current rpm estimate

$$\hat{k}(t) = \hat{k}_{rpm} r\widehat{pm}(t).$$

3 Empirical Results

We demonstrate the performance of the developed method by an example in which we emulate the work of the algorithm in off-line mode as applied to the real data measurements.

3.1 Experimental Setup and Data Specifications

The data analyzed in this example were recorded at VTT Jyväskylä laboratory scale reactor during pilot tests. It consists of 25264 samples. We used measurements of 7 variables related to our study, one of which is reported to be the output and 6 are marked as input/process value (pv) (see Table 1). The data were sampled at frequency rate 1 Hz. The purpose of the data was to study the effect of process variables on the O2 concentration and to build the corresponding model of O2 content dynamics. The data consists of a series of step response experiments designed for system identification purposes. One of the basic tasks in CFB control is to keep O2 content as much stable as possible, hence, the need for the model. In this example we intend to demonstrate how the model can improve by using the real-time estimated mass flow compared to a situation when only rpm measurements were used.

3.2 Comparison of Models

To build the model one has to solve the system identification task. One of the common approaches is to derive the transfer functions for all input/output pairs from the step response experiments [4]. We assumed that the transfer functions were of first

Table 1. Data nomenclature

Type	Description of the measurement	Sensor code	Units
OUTPUT	Oxygen content of flue gas	AIA 1.6	%
INPUT, pv	Velocity of fuel screw	SIC 2	l/min
INPUT, pv	Mass of fuel silo	WIA 2	g
INPUT, pv	Primary air flow	FICZA 1	NI/min
INPUT, pv	Secondary air flow	FICZA 3	NI/min
INPUT, pv	Secondary air flow	FICZA 4	NI/min
INPUT, pv	Secondary air flow	FICZA 5	NI/min

order and used classical geometry-based approach for identifying transfer function parameters.

However, there is no step response experiment for the fuel mass measurements from sensor WIA 2. It is clear that the mass of the fuel in the fuel tank does not have itself direct effect on the oxygen concentration, but the rate of the mass flow from the tank to the boiler has.

If the fuel were homogenous, then the feeding screw rpm measurements would completely determine the velocity of the fuel feeding. However, there are also instantaneous effects resulting from the fuel heterogeneity and discussed in the Introduction. These latter effects can only be estimated from the fuel mass measurements. Therefore, the two measurements – feeding screw rpm and fuel mass – influence the oxygen level indirectly through defining the mass flow intensity, and thus, they must be used to extract one combined signal that approximates the mass flow and is used as an input to the model.

The transfer function corresponding to the mass flow velocity for a fixed density and reactivity of the fuel can be adapted from the transfer function of the feeding screw rpm by merely modifying the gain element. The transfer function for the feeding screw rpm can be estimated from the step response experiment assuming that the fuel is homogenous during the step response experiment.

The presented models should not be considered as completely valid equivalents of the real physical processes. Rather these simplified models should be seen as local approximations of the system in a relatively small and bounded region of the system's state space. One should also keep in mind that the main purpose was to demonstrate the principal possibility to model process of O₂ content in a better and more accurate way, when the mass flow is estimated more accurately.

First, we model the oxygen concentration based on all available inputs except for the fuel mass measurement WIA2 (see Fig. 2 and Fig. 3). Although on a large scale the fitting is rather accurate, the smaller and faster phenomena are not given enough attention in this model.

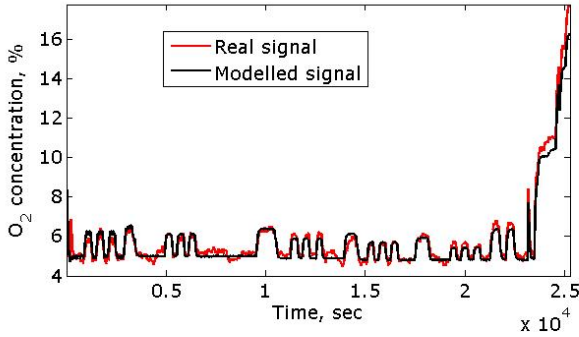


Fig. 2. Comparison of the real and modeled O2 concentration signals. WIA2 measurements are not used.

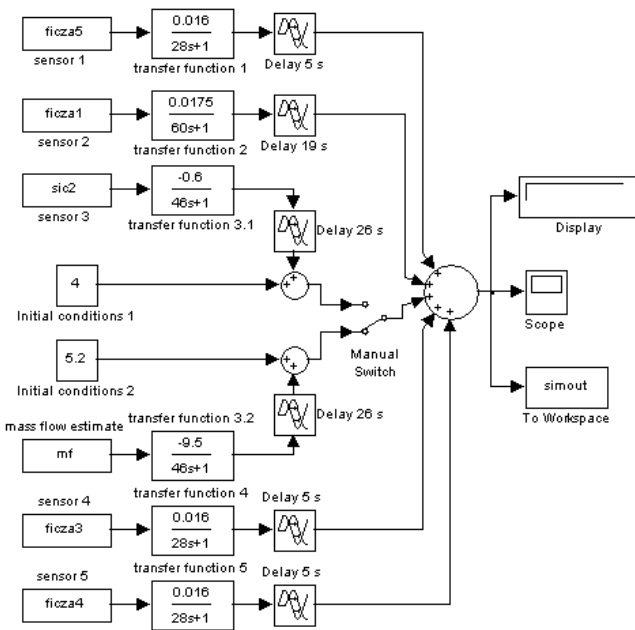


Fig. 3. Simulink model of the O2 concentration process when (1) WIA2 measurements are not used and (2) the estimated mass flow signal is used instead of the WIA2 and SIC2 measurements. Alternatives are switched by manual switch block.

Next we model the O2 concentration signal based on all inputs, but WIA2 fuel mass and SIC2 feeding screw rpm measurements were replaced by the estimated mass flow signal (see Fig. 3 and Fig. 4). One can see that the model captures now the detailed behavior fairly well.

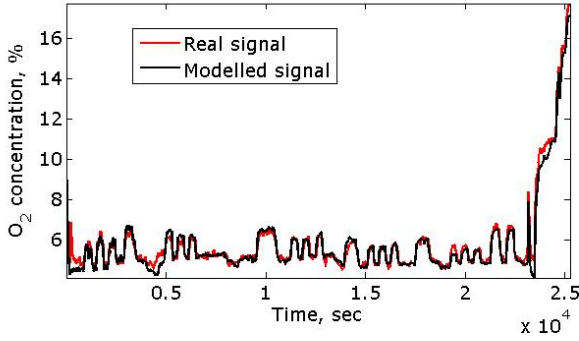


Fig. 4. Comparison of the real and modeled O₂ concentration signals. Estimated mass flow signal is used instead of SIC2 and WIA2 measurements.

The mass flow estimation was accomplished using the proposed algorithm in an off-line mode. The values of the parameters were set to the following: $N_k^{max} = 100$, $\Delta t_{min} = 1$, $N_{rpm}^{max} = 10$. The results of the estimation can be seen in Fig. 5. Almost seven hours experiment was simulated in less than two seconds. Thus, the proposed method offers more than 12000 times as faster performance as the minimal/critical time (one time sample) under the assumption that no other computational tasks are done during the same time.

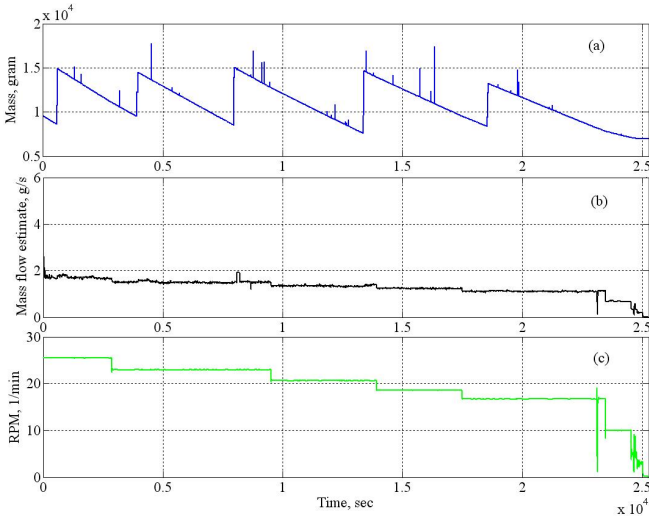


Fig. 5. Example of mass flow estimation. (a) Measurements of fuel mass, (b) Estimated mass flow, (c) Measurements of the feeding screw rpm.

4 Conclusions

The proposed algorithm demonstrated high performance in both aspects - accuracy and speed - as applied for O₂ content modeling. It was accurate enough to allow

model to capture delicate/fine behavior of the O₂ content dynamics and offered truly online performance.

The algorithm was implemented in the control system of the experimental lab-scale CFB boiler in VTT Jyväskylä.

References

1. Bakker, J., Pechenizkiy, M., Žliobaite, I., Ivannikov, A., Kärkkäinen, T.: Handling outliers and concept drift in online mass flow prediction in CFB boilers. In: Proceedings KDD Workshop on Knowledge Discovery from Sensor Data, pp. 13–22 (2009)
2. Ivannikov, A., Pechenizkiy, M., Bakker, J., Leino, T., Jegeroff, M., Kärkkäinen, T., Äyrämö, S.: Online Mass Flow Prediction in CFB Boilers. In: Perner, P. (ed.) ICDM 2009. LNCS, vol. 5633, pp. 206–219. Springer, Heidelberg (2009)
3. Pechenizkiy, M., Bakker, J., Žliobaite, I., Ivannikov, A., Kärkkäinen, T.: Online Mass Flow Prediction in CFB Boilers with Explicit Detection of Sudden Concept Drift. SIGKDD Exploration 11(2), 109–116 (2009)
4. Soderstrom, T., Stoica, P.: System identification. Prentice-Hall, Englewood Cliffs (1989)

Representation in Case-Based Reasoning Applied to Control Reconfiguration

Ons Lejri and Moncef Tagina

National School of Computer Sciences (ENSI),
SOIE Laboratory

Campus universitaire de La Manouba - 2010 La Manouba - Tunisia
{ons.lejri,moncef.tagina}@ensi.rnu.tn

Abstract. Case-Based Reasoning (CBR) is based on the use of previous experiences to solve new problems. In this work, we propose to use CBR paradigm for solving control reconfiguration problems. The reconfiguration task aims to maintain the system working despite some situations that may affect it (faults, change in production strategy, ...). The main issue is then to find the new laws or rules to use in each different situation. In this article, we especially focus on the representation part. In fact, representation is a very important task in this type of problematic as the structure of the case will affect all the other phases of the CBR cycle.

Keywords: Case-Based Reasoning (CBR), Reconfiguration, Representation, Case.

1 Introduction

Case-Based Reasoning (CBR) is a paradigm which arises from analogical reasoning and dynamic memory theory. It was inspired by Marvin Minsky's work [12] and developed by Roger Schank [15]. CBR is based on the use of previous experiences to solve new problems. Past experiences are organized on cases stocked in a case base. When a new problem is posed, the CBR system try to find similar past situations that could be helpful to solve this new problem.

The CBR cycle [1], also called the 4-Rs cycle, is composed of four different phases: Retrieving, Reusing, Revising and Retaining. The first one consists in finding in past experiences the cases whose problem is considered to be similar to the problem we are resolving. The second phase reuses extracted cases by copying or integrating them. The Revising phase adapts the retrieved solutions to suit the new problem. And the retaining phase consists in memorizing the new case obtained after resolving the problem and stocking it in the case base. This cycle is based principally on the case representation which affects each of the four phases of the CBR cycle. Therefore, the representation is a very important task to deal with while using a CBR system.

Case-Based Reasoning has been used in different types of problems where past experiences are important such as diagnosis [4], planning problems [3] and

image processing [14]. In this work, we propose to use CBR paradigm for solving control reconfiguration problems.

2 Control Reconfiguration

The reconfiguration [16,10] is a very important task related to fault tolerance. The aim of this task is to maintain the system working in all situations especially when problems occur. This includes a work of *correction* in case of fault detection and a work of *adaptation* of the control strategy when no fault is noticed (If a new production strategy is used or new requirements are adopted).

Many reconfiguration methods can be found in literature, but all these methods are based on two main ideas [7]. The first class of methods is model-based. In these methods a fault diagnosis is reached using a model-based diagnostic method. When this is done, a reconfiguration strategy is chosen in a data base containing a list of strategies and is applied to the system laws. The second class of reconfiguration methods is based on considering the system behavior as a combination of a set of elementary behaviors or states and when a fault is diagnosed, we go through the tree of the system states to reach the goal state (safe state).

The problem in these two types of methods is that we consider having an exhaustive knowledge of the system (problems that can occur as well as solutions for each of these problems); which is not always easy especially when dealing with complex system such as manufacturing systems.

Hence, we propose a reconfiguration task based on case-based reasoning. The use of CBR, will allow considering and finding solutions for new problematic situations.

In this paper, we are especially focused on case representation as it is a very important task in this type of problematic as the structure of the case will affect all the other phases of the CBR cycle.

3 Advantages of CBR for the Reconfiguration Task

CBR has many advantages that may be useful in different domains [13]. As regards our work, we consider that many characteristics of CBR are interesting for a reconfiguration task. In fact, when dealing with control reconfiguration, especially for complex systems [9,2], it is very difficult to construct an exhaustive model for reconfiguration (a rule-based model for example) as we assume that we can not predict a complete list of all situations (all faults that may occur, all strategies the system may switch to, ...) the system can reach. Thus, the use of CBR can provide us flexibility in modeling as it is no longer necessary to model all the situations and to have a complete knowledge of the domain and the system evolutivity. This reduces the knowledge extraction phase and allows us to reason with incomplete or imprecise data. Besides, CBR allows to learn over time and from past mistakes which can be beneficial when controlling complex systems.

4 Adequacy of CRB for Reconfiguration

As we saw in the previous section (*cf.* section 3), CBR has many advantages that are useful for a reconfiguration task. But till what extent can CBR be adequate for such task?

In fact, CBR, though very powerful and interesting paradigm, is not necessarily the most appropriate solution for every type of problems [13]. The type of problem we are dealing with, the domain and the problem characteristics are very important factors to take into consideration when deciding whether using or not CBR for a specific problem.

Kolodner [5,6] propose a series of five questions to which answers may help to determine the contribution that can have CBR for a problem.

1. Does the domain have an underlying model?
2. Are there exceptions and novel cases?
3. Do cases recur?
4. Is there significant benefit in adapting past solutions?
5. Are relevant previous cases obtainable?

The main issue of reconfiguration is finding the right strategy for each encountered situation by an evolving system. The systems we are dealing with are complex systems. The behaviors of such systems are not always predictable (difficulties to have an exhaustive list of all faults: example a leak in a tank that can be localized in multiple emplacement, future production strategies that can not be predicted at the current time, ...). As result, new requirements can be observed over time, making the construction of a complete domain model impossible and the use of CBR interesting. At the same time, even though the faults are different, there can be some similarities (the same type of fault but different localization, different faults with same consequences, ...) which make the use of CBR adaptation techniques and past experiences useful. All these reasons show that CBR is applicable and interesting for a reconfiguration task.

5 Representation in Case-Based Reasoning

A case is the elementary component in a CBR system. It is a capture or record of the past experience. In [6], Kolodner define a case as " *a contextualized piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoner*".

Generally speaking, a case is divided in two parts: the problem specification and the solution. Each of these two parts is defined according to the domain and the type of problem we are dealing with and the type of data we are manipulating.

As a case is the basic component of a CBR system, its representation is an important step in the process of creation of such a system. The representation has to respect the nature of the manipulated system by defining all variables and measures essential to an exhaustive definition of each particular state of the

system. Furthermore, the representation has to be convenient for extraction and reuse.

Thus, the first step when creating a CBR system is the definition of a case representation. For this, it is important to define a specification of the system and its different characteristics, and to use the available knowledge.

Remark In this work, we are interested in using CBR for a reconfiguration purpose. The diagnosis¹ phase is then supposed achieved. We consider also that the knowledge resulting of this phase (diagnosis) is data that can and will be used when defining a particular state of the system.

6 Proposition of a Case Representation

As mentioned earlier, this work is interested in representation in CBR systems aiming to propose a reconfiguration strategy (laws). In this section, we propose a case representation; this representation has to take into consideration the particularity of the system manipulated and at the same time the purpose of the use of CBR which is reconfiguration.

A case is globally divided in two parts (Fig. 1):

1. The problem representation
2. The solution representation

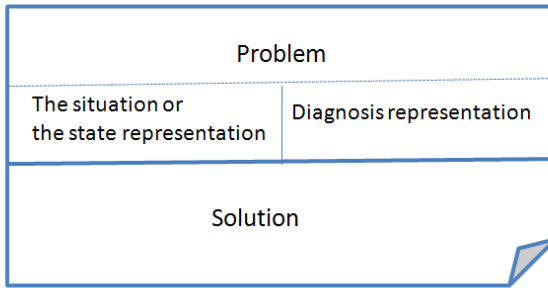


Fig. 1. Case Representation

6.1 Problem Representation

When dealing with a control reconfiguration task, knowledge from diagnosis phase is very important and necessary to work with. Thus, the diagnosis results and information has to be part of the case stocked in case base. We propose to divide the problem representation in two different parts:

¹ The diagnosis phase consist in detecting the failure of the system, locating them and eventually finding their causes.

1. The situation representation: in this part, the system state is defined. It describes the situation the system has reached and thus allows to understand the treated case.
2. The diagnosis representation: this part describes the system failures which will allow later (using also the system state and the different component states) to solve the problem.

The Situation. The problem or situation representation consists in defining the general state of the system and each of its components. A study of the system of which we are reconfiguring the control has to be done to determine the different components and the different states or situations they can present. A simple way to represent the situation is then a set of attribute-value pairs.

The Diagnosis. As mentioned previously, the diagnosis part is supposed fulfilled and thus the diagnosis result is a knowledge that can be used to lead to a solution to our reconfiguration problem. This knowledge may be essential in some cases to determine the right reconfiguration strategy we have to adopt.

The diagnosis representation will consist in the description of the diagnosis results and the failures the system is facing in each case. In our case, it is considered as additional information that allows to have a complete vision of the problem.

6.2 Solution Representation

The solution representation is the modeling of the answer for a problematic situation the system has reached. Each case stocked in the case base and expressing a previous experience has to have a solution depending on the nature of the problem treated. The solution part will eventually help to solve new problems by adapting a previous solution to the current problem.

7 A Case Study

As we are dealing with a reconfiguration problem, we will consider as example the three-tank problem (Fig 2) [9,7]. The three-tank system is the benchmark system for works on hybrid systems reconfiguration and for complex system generally speaking.

This system can present failures or new requirements that may in certain cases, and if there are no reconfiguration laws to avoid it, lead to the shutdown of this system. A simple rule-based system can not be adequate for such a problem. In fact, it is quite impossible to have an exhaustive list of all failures and future requirements (new objectives) of the system. That's why the use of CBR can be an interesting solution for this problem. In fact as mentioned earlier, the use of CBR offer a flexible way of dealing with such problems.

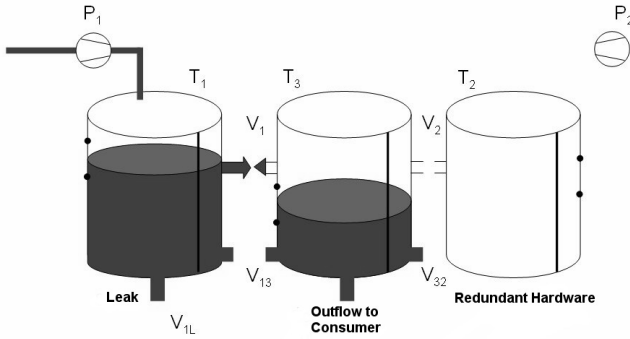


Fig. 2. The three-tank system

In this part we present the case format or representation for our particular system.

This system is characterized by a set of components (C) which defines the state of the system $C = \{P_1, P_2, V_1, V_2, V_{13}, V_{32}, C_{h_1}, C_{h_2}, C_{h_3}\}$

P_i is the pump supplying the Tank i

V_i is the valve i

C_{h_i} is the sensor measuring the liquid level h_i in Tank i

In terms of quantities this set corresponds to the set Q

$$Q = \{Q_{max}^{P1}, Q_{max}^{P2}, Q_{13}^{V1}, Q_{23}^{V2}, Q_{13}^{V_{13}}, Q_{23}^{V_{23}}, h_1, h_2, h_3\}$$

Q is representing the liquid flow

The problem representation is, thereby, defined by the expression of the different states of the different components of the system (cf. Table II). This will be expressed by a set of attribute-value pairs reflecting the actual state.

As far as the diagnosis representation is concerned, it has to express the failure or the particular abnormal situation the system is in and to transpose the effects of this situation in the system components and behavior.

The solution part has to express the laws and structural changes the system has to go through to be maintained in a working state despite the problems it is suffering from.

A distance metric has to be used to extract similar cases. The solution of the most similar case is then considered as a beginning point for resolving this new problem or the new situation the system reached.

Table 1. List of attributes defining the problem representation

Attribute	Value
P_1	opened or closed
P_2	opened or closed
V_1	opened or closed
V_2	opened or closed
V_{13}	opened or closed
V_{23}	opened or closed
V_N	opened or closed
h_1	value (liquid level in Tank1)
h_2	value (liquid level in Tank2)
h_3	value (liquid level in Tank3)
Q_1	value (supplying flow liquid for Tank1)
Q_2	value (supplying flow liquid for Tank2)
Q_{V_1}	value (liquid flow through valve V_1)
$Q_{V_{13}}$	value (liquid flow through valve V_{13})
Q_{V_2}	value (liquid flow through valve V_2)
$Q_{V_{23}}$	value (liquid flow through valve V_{23})

8 Conclusion

Case-based reasoning can be a very interesting solution for solving problems, especially, those which definition and requirements are evolving over time. For this type of problems using a rule-based solution can be exhausting and sometimes impossible when not enough knowledge is available.

In this work we were interested in using case-based reasoning for a reconfiguration purpose. CBR has been widely used to solve diagnosis problem [8,11] but it is also a great way of solving reconfiguration problems. So far, the use of CBR for reconfiguration hasn't been conveniently explored.

The reconfiguration task is a very important task as it aims for maintaining the system working in critical situations. Those situations can not always be predictable and yet are not always so different. The use of CBR will be a sort of compromise to use previous experience to solve these problems and gain time.

This work especially focused on case representation as it is considered as one of the most important issues in case-based reasoning. In fact, case representation is decisive to the success of CBR systems and the quality of the decision is correlated to the quality of the representation.

Thus, case representation is a very crucial and difficult task which has to consider the nature of the system as well as the domain and purpose. Furthermore, it has to facilitate the extraction and to be adapted to similarity calculation.

The representation format we proposed is adapted for reconfiguration task where the problem description is composed of the system's general state as well as the diagnosis representation. It allows the extraction of similar cases using a simple distance metric and can be a starting point for finding a solution for new encountered cases.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. In: *AICom - Artificial Intelligence Communications*, vol. 7(1), pp. 39–59 (1994)
2. Askari-Marnani, J., Heiming, B., Lunze, J.: Control reconfiguration: the cosy benchmark problem and its solution by means of a qualitative model. In: Part of Chapter 21 of the Final Report of the Project "Control of Complex Systems", cosy (2001)
3. Bergmann, R., Muñoz-Ávila, H., Veloso, M.M., Melis, E.: CBR Applied to Planning. In: Lenz, M., Bartsch-Spörl, B., Burkhard, H.-D., Wess, S. (eds.) *Case-Based Reasoning Technology. LNCS (LNAI)*, vol. 1400, pp. 169–199. Springer, Heidelberg (1998)
4. Göker, M.H., Howlett, R.J., Price, J.E.: Case-based reasoning for diagnosis applications. *The Knowledge Engineering Review* 20(03), 277–281 (2005)
5. Kolodner, J.L.: An introduction to case-based reasoning. *Artificial Intelligence Review* 6(1), 3–34 (1992)
6. Kolodner, J.L.: *Case Based Reasoning*. Morgan Kaufmann Publishers Inc. (1993)
7. Lejri, O., Tagina, M.: Hybrid reconfigurable petri nets for modelling hybrid reconfigurable manufacturing systems. *Journal of Studies on Manufacturing* 1(2–3), 75–84 (2010)
8. Lenz, M., Burkhard, H.D., Pirk, P., Auriol, E., Manago, M.: Cbr for diagnosis and decision support. *AI Communications - Special Issue on ECAI 1996 Budapest* 9(3) (September 1996)
9. Lunze, J., Askari, J., et al.: Three-tank reconfiguration control. *Control of Complex Systems*, 241–283 (2001)
10. Maciejowski, J.: Reconfiguring control systems by optimization. In: *European Control Conference, Brussels* (1997)
11. Marques, V., Farinha, J.T., Brito, A.: Cbr for diagnosis: evidence relevance and case adaptation. In: *ICCOMP 2009 Proceedings of the WSEAES 13th International Conference on Computers*. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA (2009)
12. Minsky, M.: A framework for representing knowledge. In: *The Psychology of Computer Vision*. McGraw-Hill, New York (1975)
13. Pal, S.K., Shiu, S.C.K.: *Foundations of Soft Case-based Reasoning*. Wiley series on intelligent systems. Wiley-Interscience publication, A John Wiley & Sons, Inc., New York (2004)
14. Perner, P., Holt, A., Richter, M.M.: Image processing in case-based reasoning. *The Knowledge Engineering Review* 20(03), 311–314 (2005)
15. Schank, R.C.: *Dynamic memory, a theory of reminding and learning in computers and people*, 1st edn. Cambridge University Press, New York (1982)
16. Tsuda, K., Mignone, D., Ferrari-Trecate, G., Morari, M.: Reconfiguration strategies for hybrid systems. In: *Proceedings of the American Control Conference, Arlington, VA, June 25–27*, pp. 868–873 (2001)

The Influence of Input and Output Measurement Noise on Batch-End Quality Prediction with Partial Least Squares

Jef Vanlaer, Pieter Van den Kerkhof, Geert Gins, and Jan F.M. Van Impe

BioTeC, Department of Chemical Engineering, KU Leuven

W. de Croylaan 46 PB 2423, B-3001 Heverlee (Leuven), Belgium

{jef.vanlaer,pieter.vandenkerkhof,geert.gins,jan.vanimpe}@cit.kuleuven.be

Abstract. In this paper, the influence of measurement noise on batch-end quality prediction by Partial Least Squares (PLS) is discussed. Realistic computer-generated data of an industrial process for penicillin production are used to investigate the influence of both input and output noise on model input and model order selection, and online and offline prediction of the final penicillin concentration. Techniques based on PLS show a large potential in assisting human operators in their decisions, especially for batch processes where close monitoring is required to achieve satisfactory product quality. However, many (bio)chemical companies are still reluctant to implement these monitoring techniques since, among other things, little is known about the influence of measurement noise characteristics on their performance. The results of this study indicate that PLS predictions are only slightly worsened by the presence of measurement noise. Moreover, for the considered case study, model predictions are better than offline quality measurements.

Keywords: Partial Least Squares, batch-end quality prediction, measurement noise statistics.

1 Introduction

The development of automated monitoring systems to assist human process operators in their decisions is an important challenge for the chemical and life sciences industry [13]. Chemical and biochemical production processes and plants are equipped with numerous sensors that measure various flow rates, temperatures, pressures, pH, concentrations, . . . Despite the frequent use of sensor measurements for automated low-level control (e.g., PID control for valve opening and closing), most information in these measurements remains unexploited as responding to abnormal events –one of the most important control tasks– most often remains a manual operation. Human operators investigate the information arising from sensors in the process and compare this information to measurements from previous process runs to detect a departure from normal operation. However, the size and complexity of modern interconnected process plants (e.g., the very high number of sensors) largely complicate this task.

A lot of research effort in the area of data-driven process monitoring has been directed towards fault detection using techniques based on *Principal Components Analysis* (PCA [3,8,11]). These techniques exploit the information in historical databases to detect deviations from nominal process behavior during a new process run. Techniques based on *Partial Least Squares* (PLS [5]) take process output (quality) measurements into account, which makes them suited not only for detection of process faults, but also for estimation of quality variables that are not measured online. Examples include the final quality of a batch process.

Batch processes are commonly used for the manufacture of products with a high added value (e.g., medicines, enzymes, high-performance polymers). Since the loss of a batch due to process faults is very costly, close monitoring of these processes is of utmost importance. Batch runs that deviate from normal process behavior should be detected as soon as possible so that corrective actions can be taken. However, due to their dynamic nature and the unavailability of final batch quality measurements while the process is running (e.g., batch-end product purity or concentration), monitoring and control of batch processes to achieve a satisfactory product quality is even more complicated. The use of multivariate PLS models to obtain batch-end quality predictions (e.g., [4,10,12]) offers a solution to this problem.

PLS has been developed to deal with large datasets of correlated measurements and to filter noise from these measurements. However, noise present on both online sensor measurements and offline quality measurements will never be removed completely and will hence negatively influence the predictive performance of the PLS models. In addition, the presence of measurement noise in the data has an influence on the selection of model inputs and the optimal model order. As these effects cause many industrial companies to be reluctant in implementing PLS techniques, this work aims at investigating the influence of input and output measurement noise characteristics, more specifically the standard deviation of Gaussian distributed noise, on PLS-based batch-end quality prediction. As a case study, an extensive dataset from a computer simulator for industrial penicillin production [2] is selected.

The paper is structured as follows. Section 2 provides a brief explanation of *Multivariate Partial Least Squares* modelling. Next, Section 3 explains how this technique is implemented for online batch-end quality prediction. In Section 4, the techniques for model order and input variable selection are discussed, after which Section 5 presents the selected case study. The results are shown and discussed in Section 6 and final conclusions are drawn in Section 7.

2 Multivariate Partial Least Squares Modelling

To predict the final quality of a batch process, a Multivariate Partial Least Squares (MPLS [9]) model is trained on historical data of normal process operation.

The modelling consists of two steps. In a first step, the data matrix containing the sensor measurements, which has a three-dimensional structure, is unfolded

to a two-dimensional matrix (Section 2.1). A general Partial Least Squares (PLS 5) model is constructed based on this two-dimensional data matrix in the second step, as explained in Section 2.2.

2.1 Data Matrix Unfolding

When for I batches, measurements of J different variables are available over K time points, a three-dimensional data matrix $\underline{\mathbf{X}}$ of size $I \times J \times K$ is obtained. To deal with this specific three-dimensional structure, the dimensionality of the matrix $\underline{\mathbf{X}}$ is reduced by means of *batch-wise data matrix unfolding* [8,10]. The matrix $\underline{\mathbf{X}}$ is divided in K slices of size $I \times J$ and these slices are placed side by side. This way, an unfolded data matrix \mathbf{X} of size $I \times JK$ is obtained. The technique preserves the batch direction: every row of the unfolded matrix corresponds to one complete batch. Figure 1 illustrates the procedure.

Other techniques for data matrix unfolding are available (e.g., variable-wise unfolding [14]). However, since batch-end quality is related to the complete batch history, batch-wise unfolding is used for prediction of the final product quality.

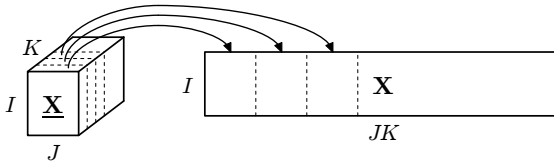


Fig. 1. Illustration of batch-wise data matrix unfolding.

2.2 Multiway Partial Least Squares (MPLS)

After data matrix unfolding, a regression model is constructed between the unfolded (input) data matrix \mathbf{X} and the (output) matrix \mathbf{Y} ($I \times L$), which contains L quality measurements for each batch in its columns, using standard two-dimensional Partial Least Squares (PLS 5).

$$\begin{cases} \mathbf{X} = \mathbf{TP}^T + \mathbf{E}_X \\ \mathbf{Y} = \mathbf{TQ}^T + \mathbf{E}_Y \end{cases} \tag{1}$$

In the PLS procedure, the input and output matrices are projected onto a lower-dimensional space, each dimension of which is defined by one of the R principal components or latent variables. These principal components are computed as linear combinations of the original measurements in such a way that they contain as much information (covariance) about the original input and output measurements as possible. The projections of \mathbf{X} and \mathbf{Y} are defined by the loading matrices \mathbf{P} ($JK \times R$) and \mathbf{Q} ($L \times R$) respectively. The scores matrix \mathbf{T} ($I \times R$)

represents the data matrices in the reduced space. The matrices $\mathbf{E}_\mathbf{X}$ and $\mathbf{E}_\mathbf{Y}$ contain the residuals or modelling errors.

The matrix \mathbf{P} is not invertible and its columns are not orthonormal. Therefore, a $JK \times R$ weight matrix \mathbf{W} with orthonormal columns is introduced to calculate the scores matrix \mathbf{T} and quality prediction \mathbf{Y} for a given measurement set \mathbf{X} . $\mathbf{P}^T \mathbf{W}$ is invertible so that the projection of the inputs \mathbf{X} on the scores space \mathbf{T} and the corresponding regression matrix \mathbf{B} ($JK \times R$) are computed as follows:

$$\mathbf{T} = \mathbf{X}\mathbf{B} \quad (2)$$

$$\mathbf{B} \triangleq \mathbf{W} (\mathbf{P}^T \mathbf{W})^{-1} . \quad (3)$$

The relation between the quality variables \mathbf{Y} and the input measurements \mathbf{X} then becomes

$$\mathbf{Y} = \mathbf{T}\mathbf{Q}^T = \mathbf{X}\mathbf{B}\mathbf{Q}^T . \quad (4)$$

3 Online Batch-End Quality Prediction

During a new batch process run, only the measurements up until the current time k are known. Missing data techniques are used to compensate for the unknown future measurements. Several techniques were investigated in [4]. The best performance was obtained with *Trimmed Scores Regression* (TSR [1]).

A major advantage of TSR is that it only requires a single PLS model to predict the batch-end quality online at every sample instance throughout the batch instead of K different models. Moreover, previous research by the authors has shown that it exhibits similar performance to the training of a new PLS model for every instance at which a prediction is asked, both for noiseless data and in industrial practice [6].

In TSR, the known part of the data matrix \mathbf{X}_{new} for a new batch (the first k columns of this data matrix, referred to as $\mathbf{X}_{\text{new},k}$) is multiplied with a matrix \mathbf{B}_k , consisting of the first k rows of the PLS regression matrix \mathbf{B} , to obtain the trimmed scores $\mathbf{T}_{\text{new},k}^*$.

$$\mathbf{T}_{\text{new},k}^* = \mathbf{X}_{\text{new},k} \mathbf{B}_k \quad (5)$$

Subsequently, a regression model is used to estimate the final scores $\mathbf{T}_{\text{new},k}$ of the new batch based on these trimmed scores. The time-varying regression matrix \mathbf{A}_k that links the estimated final scores to the trimmed scores is computed by means of a least-squares regression on the training data, for which both the complete scores $\mathbf{T}_{\text{train}}$ and the trimmed scores $\mathbf{T}_{\text{train},k}^*$ are known.

$$\begin{aligned} \mathbf{T}_{\text{train}} &= \mathbf{T}_{\text{train},k}^* \mathbf{A}_k + \mathbf{E}_\mathbf{T} \\ &\quad \downarrow \\ \mathbf{A}_k &= \left(\mathbf{T}_{\text{train},k}^{*T} \mathbf{T}_{\text{train},k}^* \right)^{-1} \mathbf{T}_{\text{train},k}^{*T} \mathbf{T}_{\text{train}} \end{aligned} \quad (6)$$

The final scores of a new batch can be estimated from the trimmed scores using this regression matrix as follows.

$$\hat{\mathbf{T}}_{\text{new},k} = \mathbf{T}_{\text{new},k}^* \mathbf{A}_k \tag{7}$$

Substituting Equation (6) into Equation (7) and exploiting the PLS relations from Equations (2) and (4), the online estimation of the batch-end quality is obtained.

$$\mathbf{Y}_{\text{new},k}^{\text{TSR}} = \mathbf{X}_{\text{new},k} \mathbf{B}_k (\mathbf{B}_k^T \mathbf{X}_{\text{tr},k}^T \mathbf{X}_{\text{tr},k} \mathbf{B}_k)^{-1} \mathbf{B}_k^T \mathbf{X}_{\text{tr},k}^T \mathbf{X}_{\text{tr}} \mathbf{B} \mathbf{Q}^T \tag{8}$$

4 Model Order and Input Variable Selection

The selection of the optimal number of principal components (i.e., the order of the PLS model) is important to obtain good predictions of the batch-end quality. Section 4.1 explains the procedure for model order selection. Moreover, a selection of the most relevant model inputs may also improve the prediction performance of the model since not all available measurements are necessarily correlated with the final batch quality. The procedure for selecting the most relevant model inputs is explained in Section 4.2.

4.1 Model Order Selection

A leave-one-out cross-validation procedure is employed to select the optimal model order R , which corresponds to the number of principal components of the PLS model. Each batch in the training dataset is left out once and MPLS models of different model orders are trained based on the other available batches. Next, the models are validated on the left out batch and the mean *Sum of Squared Errors (SSE)* over all batches in the training dataset is calculated for every model order. An *adjusted Wold’s criterion* with a threshold of 0.9, as proposed in [7], is used to select the model order. Instead of taking the number of latent variables corresponding to the observed minimum in the *SSE*-curve, the number of principal components is determined as the smallest model order R for which the following equation holds.

$$\frac{SSE(R+1)}{SSE(R)} > 0.9 \tag{9}$$

$SSE(R)$ is the (crossvalidation) *SSE* of the MPLS model with model order R . According to the adjusted Wold’s criterion, the $(R+1)^{\text{th}}$ component is only added if it significantly improves the prediction and thus decreases the crossvalidation error.

4.2 Input Variable Selection

Despite the capability of PLS models to deal with noisy data, model predictions can be improved by eliminating useless measurements that are not correlated with the final batch quality. The optimal input set is selected using a *bottom-up*

branch-and-bound procedure, assuming that the optimal set of j input variables also contains the optimal set of $j - 1$ inputs.

When J (online) measurement variables are available, J single input models are trained in a first step. Each of these models uses one of the available variables as input. The measurement variable that yields the model with the lowest leave-one-out cross-validation SSE is selected as the most important input variable. In a second step, $J - 1$ combinations of two inputs are formed by combining the first selected variable with all remaining measurements. These combinations are then used for the training of $J - 1$ two-input PLS models. Once more, the input set that results in the lowest cross-validation SSE is selected. This optimal combination of two inputs is combined with the remaining variables in the next step and the procedure continues until a ranking of all available measurements from most to least important is obtained.

Finally, a comparison is made between the cross-validation SSE for all selected input combinations. With the addition of extra input variables, the SSE will initially decrease. At a certain number of inputs however, the SSE curve reaches a minimum after which it starts rising again. The number of model inputs that corresponds to this minimum SSE value, is selected as the optimal number of input variables.

5 Case Study

Due to the need for data from a lot of batch runs with many different levels of measurement noise, a simulated process is selected as a case study. A biochemical process for penicillin fermentation at industrial scale is simulated via an extended version of the `Pensim` simulator [2]. To represent (biochemical) process variability, the initial substrate concentration, biomass concentration, and culture volume are subject to random variations for each batch. The process inputs (e.g., the substrate feed rate) exhibit variations around their setpoints as well. The process consists of two phases. Initially, the bioreactor is operated in batch mode. Once the substrate concentration drops below 0.3 g/L, the fed-batch phase is started. During this phase, additional substrate is fed into the reactor. The process is terminated after the addition of 25 L of substrate. The penicillin concentration at the end of the batch is the batch-end quality variable for which an online estimation is needed.

A total of 200 batches is simulated to investigate the influence of input and output measurement noise on the prediction of the final penicillin concentration. 15 concentrations and flows, and the temperature and pH in the bioractor are available from the simulator during the fermentation. Only 11 of these measurements are generally acquired by online sensors and thus practically available as model inputs for online prediction of the batch-end penicillin concentration. To avoid problems with badly tuned PID controllers at higher noise levels, Gaussian noise is added to the measurements of these variables after simulation. Input noise at 20 different levels is considered, which will be denoted with respect to a reference noise level described in Table [1](#).

Table 1. Overview of available online measurements with their mean nominal values and the standard deviation of the reference noise level $\sigma_{\text{noise,ref}}$ for these measurements.

Variable	Mean	$\sigma_{\text{noise,ref}}$	Variable	Mean	$\sigma_{\text{noise,ref}}$
Time [h]	-	0	Aeration rate [L/h]	8.0	$1.667e^{-1}$
DO [mmol/L]	1.1	$1.333e^{-2}$	Agitator power [W]	30.0	$3.333e^{-1}$
Volume [L]	107.5	$6.667e^{-1}$	Feed temperature [K]	296.0	$3.333e^{-1}$
pH [-]	5.0	$3.333e^{-2}$	Water flow rate [L/h]	64.2	1.667
Reactor temp. [K]	298.0	$3.333e^{-1}$	Base flow rate [L/h]	$2.5e^{-5}$	$6.667e^{-6}$
Feed rate [L/h]	0.05	$1.667e^{-3}$	Acid flow rate [L/h]	$7.9e^{-6}$	$6.667e^{-7}$

After noise addition, the measured signals are aligned and resampled to a length of 602 samples via *indicator variables*, comparable to the procedure in [2]. To obtain a monotonically increasing variable for the alignment of the batch phase, a straight line is fitted through the noisy measurements of the bioreactor volume. The time signal is added to the input measurements as an extra (aligned) variable, so that 12 online measurement signals are available for every batch. Therefore, the size of the training data matrix $\underline{\mathbf{X}}$ is $200 \times 12 \times 602$.

Output measurement noise is added to the value of the final penicillin concentration. Gaussian noise with a standard deviation of 1 to 10 percent of the mean batch-end penicillin concentration is considered. As such, measurements at 10 different levels of output noise are available.

MPLS models to predict the final penicillin concentration are constructed for all combinations of input and output noise. The optimal input variables and the model order are selected according to the procedures in Section 4 to improve the predictions. The leave-one-out cross-validation Root Mean Squared Error (*RMSE*) is calculated both offline (i.e., after conclusion of the batch operation) and online to compare the predictions at different noise levels. To assess the influence of measurement noise on quality predictions without the influence of different model inputs, the prediction performance of models that use all 12 available measurements as inputs is also compared for different input and output noise levels. All calculations are performed thrice with different noise values sampled from the respective Gaussian distributions.

6 Results and Discussion

The next sections present the results of the study. The discussion of the influence of input noise and output measurement noise on batch-end quality prediction are decoupled in Sections 6.1 and 6.2 respectively. In each part, the influence of the noise on input variable and model order selection, offline quality prediction and online quality prediction is discussed.

6.1 Input Measurement Noise

Input Variable and Model Order Selection

After the addition of input measurement noise and alignment of the data, the optimal set of input variables and the model order are selected for every input noise level according to the procedure in Section 4.

In the noiseless case, 6 inputs (Dissolved Oxygen (*DO*), feed rate, time, pH, reactor temperature, and water flow rate) are selected. Several of these variables (e.g., pH and temperature) are PID controlled and vary only slightly. When even low amounts of measurement noise are added to the data, these measurements are rendered uninformative and a lower number of inputs is selected. Up to a noise level of $1/8^{\text{th}}$ of the reference level, the selected number of inputs is mostly 3. *DO*, feed rate, and time remain the most important input variables.

At a noise level of $1/8^{\text{th}}$ of the reference level, the noise has reached the size of the normal variation of the *DO* measurements. 6 inputs are again selected in an attempt to filter out the noise by exploiting the variable correlation.

At higher noise levels, *DO* measurements become uninformative due to the noise. The reactor volume is then selected as the most important variable. The number of inputs varies between 2 and 5.

The model order shows a decreasing trend with increasing input noise level, ranging from 9 for the noiseless case to 1-2 for the highest tested noise level (6 times the reference level). Ideally, the model order is a measure for the number of independent underlying phenomena that determine the course of the process. When more noise is added to the data, more and more of these phenomena are masked and fewer latent variables are selected.

Offline Quality Prediction

Offline prediction of the batch-end quality is the estimation of –in this case– the final penicillin concentration at the end of the batch operation. When the batch operation has finished, the complete data matrix $\underline{\mathbf{X}}$ is known, so no compensation for missing variables is needed. Figure 2 shows the average offline prediction *RMSE* as a function of the input noise level when no noise is present in the output measurements for both MPLS models with optimal inputs (*full curve*) and models which employ all 12 available online measurements as input variables (*dashed curve*). The selection of optimal input variables leads to better offline estimations of the batch-end penicillin concentration, evidenced by the lower *RMSE* values. However, both curves exhibit the same trend. As expected, the *RMSE* increases (so the prediction performance decreases) with increasing input noise level. The increase is most obvious at low noise levels, while at higher noise levels, the increase is less pronounced and the *RMSE* saturates. Even at high input noise levels, the *RMSE* is still relatively small and very good quality predictions are obtained.

This is also concluded from Figure 3, which shows a plot of the offline leave-one-out cross-validation prediction against the real final penicillin concentration for both the noiseless case (Figure 3(a)) and an input noise level of 6 times the reference level (Figure 3(b)). Without measurement noise, a nearly perfect

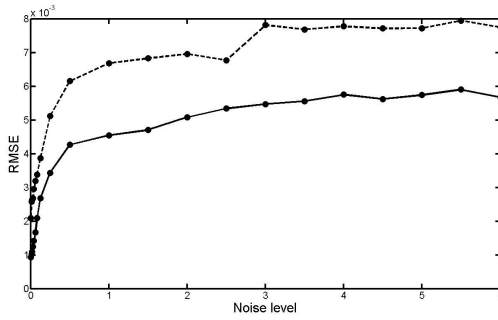


Fig. 2. Leave-one-out cross-validation $RMSE$ for offline prediction of the final penicillin concentration in function of the input noise level: with selection of model inputs (—) and with all available inputs (- -). No output measurement noise is present in the data.

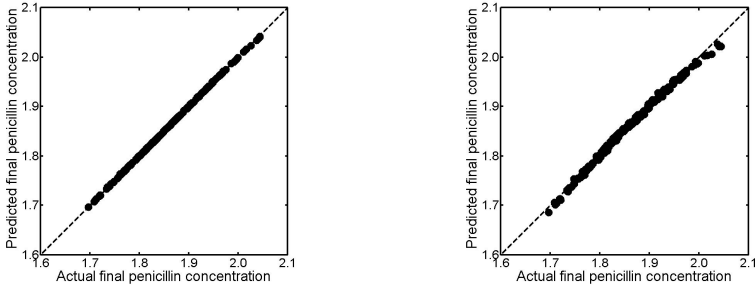


Fig. 3. Optimized offline prediction of the final penicillin concentration versus real penicillin concentration for the noiseless case (*left*) and an input noise level of 6 times the reference level (*right*). No output measurement noise is present in the data.

prediction is obtained (also evidenced by an $RMSE$ of 0.001). However, even for the highest noise level under study –much higher than the noise encountered in industry–, the estimated quality approaches the real quality very well. Hence, a very efficient removal of the input noise from the data is achieved.

Online Quality Prediction

Using Trimmed Scores Regression (TSR) to compensate for missing future measurements, online predictions of the final penicillin concentration are obtained as explained in Section 3. The evolution in time of the maximal relative deviation of the online prediction from the real final penicillin concentration is depicted in Figure 4, both for the noiseless case and for noise of 6 times the reference level.

Initially, the predicted penicillin concentration deviates considerably from the real final value for both cases, since very few measurements are available. For the noiseless case, the deviation quickly drops below 1% as the batch progresses

and by the end of the process run, the prediction has evolved towards the correct value. For noise of 6 times the reference level, the relative deviation decreases more slowly. Nonetheless, a stable prediction that deviates less than 1% from the real batch-end quality is obtained in about 200 samples.

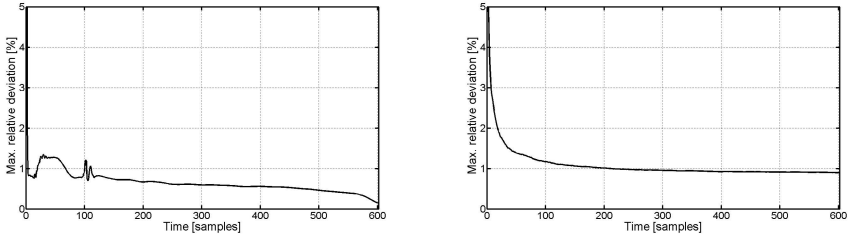


Fig. 4. Maximal relative deviation of the online prediction from the real final penicillin concentration in function of time for the noiseless case (*left*) and input measurement noise of 6 times the reference level (*right*). No output noise is present in the data.

An overview of the maximal relative prediction deviation for different input noise levels and sample times is given in Table 2(a). At first sight, the results are a little unexpected: adding noise improves the online prediction in some situations. This is especially visible during the batch phase (the first 101 samples of the process), where better predictions are obtained with noise of the reference level than at a noise level which is 16 times smaller. The selection of input variables, which aims at improving the offline batch-end quality prediction, does not necessarily guarantee optimal online predictions. Especially when the process consists of different phases, selecting one set of input variables for the complete process may result in a decrease in online prediction performance during certain phases.

As discussed earlier, different model inputs are selected for low and high noise levels. Apparently, the selected inputs for the low noise levels do not contain enough information to obtain good online predictions during the batch phase. This is evidenced by the fact that models that employ all available online measurements as inputs result in better online predictions during the batch phase for lower noise levels, as shown in Table 2(b). Consequently, it is better to use all available model inputs to obtain good online predictions from the start of the process. Another option is the training of different models (with different inputs) for all process phases.

From the results in Table 2, it can be concluded that higher input noise levels result in slightly worse prediction performance. However, the predictions improve with time and good and stable predictions are obtained in fewer than 200 samples for all noise levels.

Table 2. Influence of input measurement noise level on the maximal relative deviation of the online prediction from the real final penicillin concentration for models with (a) selected inputs and (b) all available inputs. No output noise is present in the data.

Time	No noise	Level $1/16$	Level 1	Level 3	Level 6
1	12.4%	12.5%	1.7%	4.3%	7.1%
50	1.3%	10.8%	1.0%	1.1%	1.2%
100	0.9%	7.9%	1.0%	1.0%	1.2%
200	0.7%	1.0%	0.9%	1.0%	1.0%
300	0.6%	0.7%	0.8%	0.9%	1.0%
400	0.6%	0.6%	0.8%	0.9%	0.9%
500	0.5%	0.5%	0.7%	0.9%	0.9%
602	0.1%	0.3%	0.7%	0.9%	0.9%

(a)

Time	No noise	Level $1/16$	Level 1	Level 3	Level 6
1	0.9%	2.5%	3.1%	5.0%	7.5%
50	0.9%	1.1%	1.1%	1.3%	1.6%
100	1.1%	1.1%	1.2%	1.3%	1.4%
200	0.8%	0.9%	1.2%	1.6%	1.3%
300	0.7%	0.8%	1.1%	1.5%	1.3%
400	0.6%	0.7%	1.1%	1.4%	1.2%
500	0.5%	0.6%	1.1%	1.3%	1.2%
602	0.3%	0.5%	1.0%	1.2%	1.2%

(b)

6.2 Output Measurement Noise

Input Variable and Model Order Selection

When output noise is added to measurements of the final penicillin concentration, the number of selected inputs varies greatly for different combinations of input and output noise levels. However, dissolved oxygen concentration (only at low input noise levels) or reactor volume always remain the most important variables. No real conclusions can be drawn about the importance of the other available measurements since various combinations of variables are selected at different combinations of input and output noise levels.

The optimal model order decreases quickly with the size of the output measurement noise. For output noise with a standard deviation of 1 percent of the mean final penicillin concentration 1 to 3 latent variables are selected at input noise levels smaller than the reference level. At higher input noise levels a model order of 1 is selected. For output noise with a standard deviation of 2 to 10 percent of the mean batch-end quality measurement a model order of 1 is selected in most cases.

Offline Quality Prediction

The full curve in Figure 5 gives an overview of the leave-one-out cross-validation *RMSE* for offline prediction of the batch-end penicillin concentration in function

of the output noise standard deviation at the reference input noise level when optimal input variables are selected. The course of the curve is very similar for other input noise levels and for models that employ all available input variables. The *RMSE* increases linearly with the size of the output noise and its value is approximately equal to the standard deviation of the output measurement noise. Thus, it seems that the size of the output noise has a very big influence on the prediction performance of the PLS models. However, the *RMSE* was calculated by comparing the model predictions to the final penicillin concentration measurements, which contain noise. By comparing the predictions to the real (noiseless) value of the batch-end quality an actual *RMSE* value is obtained. The dashed curve in Figure 5 shows the course of this actual *RMSE* in function of the output noise standard deviation. From this curve it becomes clear that the influence of the output noise on the offline prediction is actually very small. Unlike the measurement *RMSE*, the actual *RMSE* increases only slightly with increasing output noise size and even at an output noise standard deviation of 10 percent, the size of the actual *RMSE* is around 1 percent of the value of the final penicillin concentration.

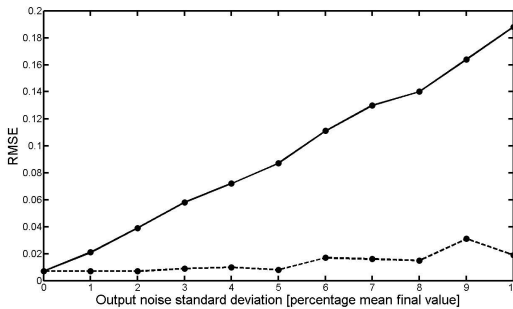


Fig. 5. Measurement (—) and actual (- -) leave-one-out cross-validation *RMSE* for offline prediction of the final penicillin concentration in function of the output noise standard deviation at the reference input noise level with model input selection

A graphical representation of this result is given in Figure 6. The graph on the left shows a plot of the measured final penicillin concentration, which contains noise with a standard deviation of 5% of the mean concentration, versus the actual (noiseless) batch-end penicillin concentration. In the middle graph, the model prediction is plotted against the measured penicillin concentration. Correlation between these variables is small and the size of the deviation of the prediction from the measurements is equal to the size of the measurement noise in the left graph. However, when the predicted penicillin concentration is plotted against the actual value in the graph on the right, a high correlation is obtained.

Of course, perfect (noiseless) quality measurements are never available in industry. However, as illustrated in this case study, PLS model predictions may

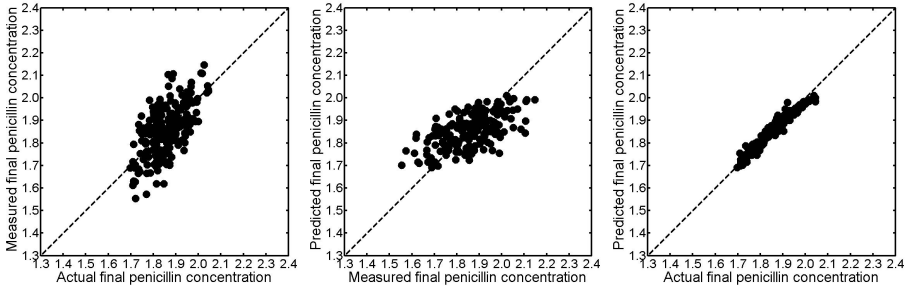


Fig. 6. Results of the offline prediction of the final penicillin concentration with optimized input variables with input noise of the reference level and Gaussian output noise with a standard deviation of 5% of the mean final penicillin concentration: measured vs. actual penicillin concentration (*left*), predicted vs. measured penicillin concentration (*middle*) and predicted vs. actual penicillin concentration (*right*).

be better than offline quality measurements, even when these noisy measurements are used to train the models. It is important to be aware of the size of the noise on the quality measurements, since even perfect predictions result in a measurement $RMSE$ of approximately the same size as the standard deviation of the measurement noise σ_{noise} . This is corroborated by the formulas of both variables:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (\hat{y}_i - y_i)^2}{N}} \tag{10}$$

$$\sigma_{\text{noise}} = \sqrt{\frac{\sum_{i=1}^N (y_{i,\text{real}} - y_i)^2}{N - 1}} \tag{11}$$

with \hat{y}_i the model prediction, y_i the measured quality and $y_{i,\text{real}}$ the real (noiseless) quality of batch i , and N the number of training batches. In case of a nearly perfect prediction ($\hat{y}_i \approx y_{i,\text{real}}$) and for a sufficiently high number of training batches, these formulas are approximately the same.

In this case, it is valuable to temporarily invest in some extra quality measurements with higher accuracy to check the prediction performance of the model.

Online Quality Prediction

As for the offline quality prediction, output measurement noise has very little influence on online batch-end quality prediction when the deviation of the prediction from the actual (noiseless) final penicillin concentration is considered. Online predictions are slightly worse than in the case where no output noise is present in the data, but good and stable predictions are still obtained within an acceptable time span.

7 Conclusions

In this paper, the influence of input and output measurement noise characteristics on PLS-based batch-end quality prediction is investigated. As a case study, realistic computer-generated data of a fed-batch process for penicillin production are used. Gaussian noise of different levels (i.e., different size of the noise standard deviation) is added to the process input and output measurements. The effect of the noise level on input and model order selection, and offline and online prediction performance is studied.

The information content of measurements decreases with increasing noise level. While measurements of controlled variables, which vary only slightly, may be informative in the noiseless case, they are soon rendered uninformative when noise is added to the data. When the size of the input noise approaches the normal variation of informative measurements, PLS is no longer able to filter out the noise and a new set of optimal input variables is selected. Since higher noise values mask more and more important underlying phenomena, the model order decreases with both the input and output noise level.

The offline prediction performance of the PLS models decreases only slightly with increasing noise levels. Even for noise levels much higher than those encountered in industry, very good offline quality predictions are obtained. This proves the ability of PLS models to filter the noise from the data. Since no perfect (noiseless) quality measurements are available in industry, it is important to be aware of the size of the measurement noise. As illustrated in the case study, model predictions may be better than the measurements since even perfect predictions result in a measurement *RMSE* of approximately the same size as the standard deviation of the noise on the quality measurement.

When the selection of different model inputs at different noise levels is not taken into account, online predictions of the batch-end quality using Trimmed Scores Regression (TSR) deteriorate slightly with increasing levels of both input and output measurement noise. Despite the slightly lower prediction performance at higher noise levels, accurate and stable online predictions are obtained, even at noise levels much higher than in industrial practice. Future research will investigate the generalization of the obtained results.

Acknowledgements. Work supported in part by Project PVF/10/002 (OPTEC Optimization in Engineering Center) of the Research Council of the KU Leuven, Project KP/09/005 (SCORES4CHEM) of the Industrial Research Council of the KU Leuven, and the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian Federal Science Policy Office. J. Vanlaer and P. Van den Kerkhof are funded by a Ph.D grant of the Agency for Innovation by Science and Technology (IWT). J. Van Impe holds the chair Safety Engineering sponsored by the Belgian chemistry and life sciences federation *essenscia*. The scientific responsibility is assumed by the authors.

References

1. Arteaga, F., Ferrer, A.: Dealing with Missing Data in MSPC: Several Methods, Different Interpretations, Some Examples. *J. Chemometr.* 16, 408–418 (2002)
2. Birol, G., Ündey, C., Çinar, A.: A Modular Simulation Package for Fed-Batch Fermentation: Penicillin Production. *Comput. Chem. Eng.* 26, 1553–1565 (2002)
3. Eriksson, L., Johansson, E., Kettaneh, N., Wold, S.: Multi- and Megavariate Data Analysis: Principles and Applications. Umetrics Academy (2002)
4. Garcia-Munoz, S., Kourti, T., MacGregor, J.: Model Predictive Monitoring for Batch Processes. *Ind. Eng. Chem. Res.* 43, 5929–5941 (2004)
5. Geladi, P., Kowalski, B.: Partial Least-Squares Regression: a Tutorial. *Anal. Chim. Acta* 185, 1–17 (1986)
6. Gins, G., Vanlaer, J., Van Impe, J.: Online Batch-End Quality Estimation: Does Laziness Pay Off? In: Quevedo, J., Escobet, T., Puig, V. (eds.) Proceedings of the 7th IFAC International Symposium on Fault Detection, Supervision and Safety of Technical Processes (Safe Process 2009), pp. 1246–1251 (2009)
7. Li, B., Morris, J., Martin, E.: Model Selection for Partial Least Squares Regression. *Chemometr. Intell. Lab.* 64, 79–89 (2002)
8. Nomikos, P., MacGregor, J.: Monitoring Batch Processes Using Multiway Principal Component Analysis. *AIChE J.* 40(8), 1361–1375 (1994)
9. Nomikos, P., MacGregor, J.: Multiway Partial Least Squares in Monitoring Batch Processes. *Chemometr. Intell. Lab.* 30, 97–108 (1995)
10. Nomikos, P., MacGregor, J.: Multivariate SPC Charts for Monitoring Batch Processes. *Technometrics* 37(1), 41–59 (1995)
11. Simoglou, A., Georgieva, P., Martin, E., Morris, A., de Azevedo, S.: On-line Monitoring of a Sugar Crystallization Process. *Comput. Chem. Eng.* 29(6), 1411–1422 (2005)
12. Ündey, C., Ertunç, S., Çinar, A.: Online Batch/Fed-Batch Process Performance Monitoring, Quality Prediction, and Variable-Contribution Analysis for Diagnosis. *Ind. Eng. Chem. Res.* 42, 4645–4658 (2003)
13. Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.: A Review of Process Fault Detection and Diagnosis. Part I: Quantitative Model-Based Methods. *Comput. Chem. Eng.* 27, 293–311 (2003)
14. Wold, S., Geladi, P., Ebensen, K., Öhman, J.: Multi-way Principal Components- and PLS-Analysis. *J. Chemometr.* 1(1), 41–56 (1987)

An Evolving Associative Classifier for Incomplete Database

Kaoru Shimada

Fukuoka Dental College, 2-15-1, Tamura, Sawara, Fukuoka, 814-0193, Japan
shimada@college.fdcnet.ac.jp

Abstract. An associative classification method for incomplete database is proposed based on an evolutionary rule extraction method. The method can extract class association rules directly from the database including missing values and build an associative classifier. Instances including missing values are classified by the classifier. In addition, an evolving associative classifier is proposed. The proposed method evolves the classifier using the labeled instances by itself as acquired information. The performance of the classification was evaluated using artificial incomplete data set. The results showed that the proposed evolving associative classifier has a potential to expand the target data for classification through its evolutionary process and gather useful information itself.

Keywords: classification, association rule, incomplete data, evolutionary computation.

1 Introduction

Association rule mining is the discovery of association relationships or correlations among a set of attributes (items) in a database [1]. Association rule in the form of ‘If X then Y ($X \rightarrow Y$)’ is interpreted as ‘the set of attributes X are likely to satisfy the set of attributes Y ’. When the right hand side of the rule is the class label, it can be used for classification. Associative classification techniques [2,3,4,5] have been proposed which have achieved quite effective performance. These methods first mine class association rules (CARs) from training data, and then build a classifier using these rules. However, previous approaches cannot handle incomplete database including missing values in some instances.

Generally, conventional rule extraction methods regard the database as complete, or disregard instances including missing values. Specifically, instances including missing values are deleted for rule mining or filled in with the mean values or frequent category [6,7]. When the data set has a huge number of instances and missing rate is low, it is easy to take these policies, however, the data mining such as in the medical science fields is different from the situation. Usually, the number of instances in the medical science data is not so large. Data sets probably include many missing values caused by the failure of experiments or the lack of personal information. In these cases, it is not possible to fill in the missing values using above way. In addition, classification for incomplete

data also should be considered. In the conventional associative classifier, the first matching rule usually makes the prediction. Therefore, the order of the rules in the classifier affects the accuracy of the classification. In the case that missing values exist in both of the training and testing data, the multiple matched rules based method could be appropriate.

Recently, association rule mining tool for incomplete data set has been proposed using an evolutionary computation method [8]. The tool uses the basic structure of Genetic Network Programming (GNP) [9] and adopts a new strategy in evolution to execute tasks through generations. GNP based method extracts association rules without filling in the missing values. In [8], general type association rules ($X \rightarrow Y$) for the analysis of given data set was focused. In this paper, we focus a CARs mining method to the classification problems for incomplete database. The conventional GNP based classification method [10,11] is applicable only for complete database. In addition, a new method for building an evolving classifier is proposed. In GNP based rule extraction method, rules satisfying the given conditions are stored in a rule pool through GNP generations. GNP individuals evolve in order to store new interesting rules into the pool as many as possible, not to obtain the individual with highest fitness value. Therefore, the GNP based method can quit the rule extraction anytime when enough number of rules are obtained for building a classifier. Applying extracted rules at the moment for classification, we can obtain new labeled instances. If we join these just labeled data into training data, then extended training data can be constructed. We can repeat this process and evolve the classifier using acquired information. This mechanism can expand the target data for classification through its evolutionary process and gather useful information.

This paper is organized as follows. In the next section, some related concepts and explanations on CARs and classification are presented. A method of CARs extraction from an incomplete database is described in Section 3. In Section 4, a new algorithm for evolving classifier is introduced. Experimental results are presented in Section 5, and conclusions are given in Section 6.

2 Rules and Classification

2.1 Class Association Rules in Incomplete Database

Let A_i be an attribute (item) in the database. In order to describe the algorithm clear, we indicate the attribute values of the instances by 1 or 0 as shown in Table 1 [8]. In addition, missing values are indicated as ‘ m ’. This means that the absence of item A_i is described as $A_i = 0$ and lack of information of A_i is indicated as ‘ $A_i = m$ ’. For example, $ID = 3$ in Table 1 misses the value of attribute A_4 . The meaning of this is like that item A_4 was sold out in $ID = 3$, then we cannot know whether the customer bought A_4 or not. In this paper, we define missing rate as the ratio of the number of missing values and the total number of attribute values. For example, 8 missing values are found within 32 values of A_1, A_2, A_3 and A_4 in Table 1. In this case, missing rate is $8/32=25\%$.

Table 1. An example of incomplete database

<i>ID</i>	<i>A</i> ₁	<i>A</i> ₂	<i>A</i> ₃	<i>A</i> ₄	<i>C</i>	<i>A</i> ₁ ∧ <i>A</i> ₂ ∧ <i>A</i> ₃ → (<i>C</i> =1)
1	1	1	1	0	1	satisfy (available)
2	1	1	1	1	0	not satisfy (available)
3	1	1	1	<i>m</i>	1	satisfy (available)
4	1	1	<i>m</i>	0	0	not satisfy (available)
5	0	1	<i>m</i>	1	1	not satisfy (available)
6	0	<i>m</i>	1	1	1	not satisfy (available)
7	1	1	<i>m</i>	0	1	cannot judge (unavailable)
8	<i>m</i>	<i>m</i>	1	<i>m</i>	1	cannot judge (unavailable)

Let *C* be the class label and the database has no missing class labels. When the data has *K* classes, the class labels are denoted as *C* = *k* (*k* = 0, 1, . . . , *K* − 1). In addition, *X* denotes the combination of attributes like *X* = (*A*_{*j*} = 1) ∧ . . . ∧ (*A*_{*k*} = 1). *X* is represented briefly as *A*_{*j*} ∧ . . . ∧ *A*_{*k*}. Let *N* be the number of available instances for the rule measurements. If the number of instances containing *X* in the database equals α , then we define $support(X) = \alpha/N$. β and γ are used as the number of instances labeled (*C* = *k*) and *X* ∧ (*C* = *k*), respectively. The rule *X* → (*C* = *k*) has measures defined by the following:

$$support(X \rightarrow (C = k)) = \frac{\gamma}{N}, \quad confidence(X \rightarrow (C = k)) = \frac{\gamma}{\alpha},$$

$$support(C = k) = \frac{\beta}{N}, \quad \chi^2(X \rightarrow (C = k)) = \frac{N(N \cdot \gamma - \alpha\beta)^2}{\alpha\beta(N - \alpha)(N - \beta)}.$$

The number of instances for the calculation of measurement is different rule by rule [8]. We demonstrate the feature of the available instances for the rule measurements using Table 1. Let (*A*₁ = 1) ∧ (*A*₂ = 1) ∧ (*A*₃ = 1) → (*C* = 1) be a candidate rule. The instance *ID* = 3 satisfies this rule even though value for *A*₄ is missed. When at least one of the attribute values of *A*₁, *A*₂ or *A*₃ equal 0, the instance does not satisfy the rule. *ID* = 5 and 6 are available to judge for the rule even if they have missing values. These instances are available for the calculation of rule measurements. *ID* = 7 and 8 are unavailable, because we cannot judge whether the instances satisfy the rule or not by missing values. Measurements of the above rule are as follows:

$$support(A_1 \wedge A_2 \wedge A_3 \rightarrow (C=1)) = \frac{2}{6}, \quad confidence(A_1 \wedge A_2 \wedge A_3 \rightarrow (C=1)) = \frac{2}{3}.$$

ID = 4 is available for the above rule, however it is unavailable for (*A*₁ = 1) ∧ (*A*₂ = 1) ∧ (*A*₃ = 1) → (*C* = 0). We should exclude the instances whose attribute values in a candidate rule equal 1 or *m* except the case all the attribute values equal 1.

2.2 Building a Multi Rules Based Classifier

Rule-based classification usually involves two stages: training and testing. In the training stage, important CARs are generated for the classification. In the testing

stage, obtained rules are applied to estimate of the test data. The proportion of predicting the test data correctly is called the accuracy of classification. There are roughly two types of models for building classifiers based on CARs: ordered rule based and unordered rule based. In the ordered rule based model, CARs are reordered by confidence and support, then, the first matching rule usually makes the prediction [2]. On the other hand, unordered rule based model compares the accuracy or score of all classes obtained from the multiple matched rules. The class having the highest accuracy or score is used for the prediction.

In this paper, we define the important CARs as satisfying the following:

$$\text{support}(X \rightarrow (C = k)) \geq \text{supp}_{min}, \quad (1)$$

$$\chi^2(X \rightarrow (C = k)) > \chi^2_{min}, \quad (2)$$

$$\text{confidence}(X \rightarrow (C = k)) \geq \text{support}(C = k), \quad (3)$$

where, supp_{min} and χ^2_{min} are the minimum support and χ^2 values given by users in advance. (3) is required for the positive association for (2). For example, instances in the medical field sometimes include different characteristics individual by individual. Therefore, important rules do not always have high confidence values. In the classification method based on the matched multiple rules, we can say that it is recommended to use not only the strict rules having high confidence value, but also the rules having high χ^2 value even if they have a low confidence value. The method described in [10] for building a classifier is extended to the incomplete data set as follows. *available rule* is defined as the rule which can judge whether the new instance satisfies the antecedent of the rule or not.

[Input] A set of CARs and an instance to be classified

[Output] Class predicted by the classifier

[Method] 1. *available(k)*: compute the total number of available rules satisfying $C=k$ in the classifier ($k=0, 1, \dots, K-1$)

2. *match(k)*: compute the number of rules in the classifier, whose antecedent match the new data and satisfy $C=k$

3. $\text{score}(k) = \frac{\text{match}(k)}{\text{available}(k)}$

If *available(k)* = 0 then *score(k)* = 0

4. the instance is predicted as $C = \arg \max \text{score}(k)$

3 Rule Mining Method

3.1 Genetic Network Programming

GNP-based rule mining for incomplete database is reviewed briefly [8]. GNP is an evolutionary method, which uses the directed graph structure [9]. A given number of GNP individuals form a population and evolve toward to a given purpose. GNP individual is composed of two kinds of nodes: judgment node and processing node. Judgment nodes are the set of J_1, J_2, \dots , which work as *if-then* type decision making functions. Processing nodes are the set of P_1, P_2, \dots , which work as some kind of action/processing functions. The practical roles of these

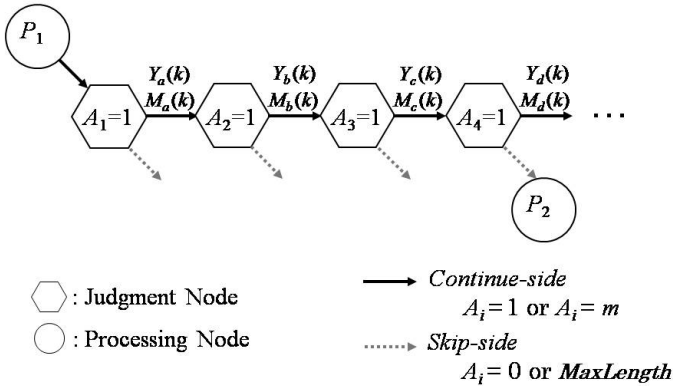


Fig. 1. An example of node connection of GNP

nodes are predefined and stored in the library by supervisors. The execution of GNP starts from the start node, and the next node to be executed is determined according to the connection and judgment result of the current activated node.

All individuals in a population have the same number of nodes. As the genetic operators for GNP, crossover and mutation are used. Crossover operator affects two parent individuals. All the connections or contents of the uniformly selected corresponding nodes in two parents are swapped by crossover rate P_c . Mutation operator affects one individual. All the connections of each node or node contents are changed randomly by mutation rate P_m .

3.2 Basic Ideas of Rule Representation

Attributes and their values correspond to the functions of judgment nodes in GNP. The connections of nodes are represented as association rules. Fig. 1 shows a sample of the connection of nodes in a part of GNP [8]. P_1 is a processing node and is a starting point of rule representation. ‘ $A_1 = 1$ ’, ‘ $A_2 = 1$ ’, ‘ $A_3 = 1$ ’ and ‘ $A_4 = 1$ ’ in Fig. 1 denote the functions of judgment nodes. The connections of these nodes represent CARs, for example, $(A_1 = 1) \rightarrow (C = k)$, $(A_1 = 1) \wedge (A_2 = 1) \rightarrow (C = k)$ and $(A_1 = 1) \wedge (A_2 = 1) \wedge (A_3 = 1) \rightarrow (C = k)$.

If some of the rules represented in Fig.1 are interesting, then rules symbolized by after changing the connections or contents of nodes could be candidates of interesting ones. We can obtain these rule candidates effectively by GNP genetic operations. In the next GNP generation, the candidates will be examined.

Fig. 2 shows a basic structure of GNP for rule extraction. Each processing node has an inherent numeric order (P_1, P_2, \dots, P_s) and is connected to a judgment node. Each processing node points the first judgment node to interpret the rules like P_1 in Fig. 1. Each judgment node has two connections: Continue-side and Skip-side. The Continue-side of the judgment node is connected to another judgment node. The Skip-side of the judgment node is connected to the next numbered processing node. In Fig. 2, the Skip-side of judgment nodes

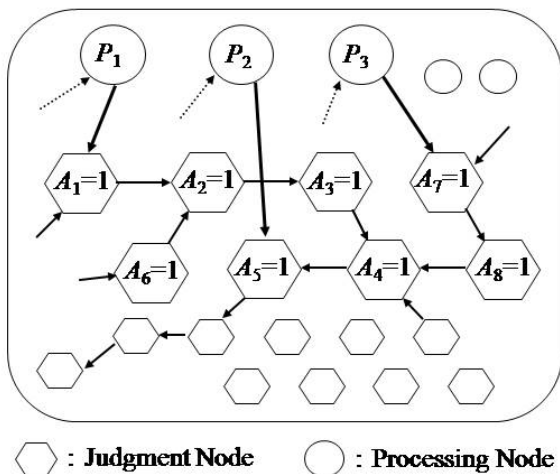


Fig. 2. An example of node connection in a GNP individual

are abbreviated. Judgment nodes can be reused and shared with some other rule representations because of the GNP’s feature. Therefore, many rules can be considered using this structure. Start node connects to P_1 .

GNP examines the attribute values of each instance using judgment nodes. Judgment node determines the next node by a judgment result. When the attribute value equals 1, then we move to the Continue-side. In the case that the attribute value equals 0, the Skip-side is used for the transition. For example, in Table 1, the instance $1 \in ID$ satisfies $A_1 = 1$, $A_2 = 1$, $A_3 = 1$ and $A_4 = 0$, therefore, the node transition from P_1 to P_2 occurs in Fig. 1. When attribute value is missing, then, move to the Continue-side. If the transition to Continue-side connection continues and the number of the judgment nodes from the processing node becomes a cutoff value ($MaxLength$), then, the connection is transferred to the next processing node using the Skip-side obligatorily.

Skip-side of the judgment node is connected to the next numbered processing node. Then, another examination of attribute values starts at next processing node. If the examination of attribute values from the starting point P_s ends, then GNP examines the instance $2 \in ID$ from P_1 likewise. Thus, all instances in the database are examined.

3.3 Calculation of Rule Measurements

The numbers of instances moved to the Continue-side at each judgment node are counted up and stored in memories. Each judgment node also examines the case of $C = k$ at the same time. In Fig. 1, $Y_a(k)$, $Y_b(k)$, $Y_c(k)$ and $Y_d(k)$ are the numbers of instances which belong to class $C = k$ and move to the Continue-side at each judgment node satisfying that all the attribute values are equal to 1 from the processing node (Y value). $M_a(k)$, $M_b(k)$, $M_c(k)$ and $M_d(k)$ are the

Table 2. Measurements of class association rules

Class association rule	support	confidence
$A_1 \rightarrow (C = k)$	$\frac{Y_a(k)}{\sum_{i=0}^{K-1} N_a(i)}$	$\frac{Y_a(k)}{\sum_{i=0}^{K-1} Y_a(i)}$
$A_1 \wedge A_2 \rightarrow (C = k)$	$\frac{Y_b(k)}{\sum_{i=0}^{K-1} N_b(i)}$	$\frac{Y_b(k)}{\sum_{i=0}^{K-1} Y_b(i)}$
$A_1 \wedge A_2 \wedge A_3 \rightarrow (C = k)$	$\frac{Y_c(k)}{\sum_{i=0}^{K-1} N_c(i)}$	$\frac{Y_c(k)}{\sum_{i=0}^{K-1} Y_c(i)}$
$A_1 \wedge A_2 \wedge A_3 \wedge A_4 \rightarrow (C = k)$	$\frac{Y_d(k)}{\sum_{i=0}^{K-1} N_d(i)}$	$\frac{Y_d(k)}{\sum_{i=0}^{K-1} Y_d(i)}$

number of instances at each judgment node satisfying that the attribute values are equal to 1 or missing values. The number of available instances for the rule measurements calculation are obtained as follows: $N_x(k) = N_T - (M_x(k) - Y_x(k))$, where, N_T is the total number of instances in the database.

Measurements of rules are calculated as follows using the above numbers. For example, in Fig. 1, $Y_d(k)$ indicates the number of instances satisfying $A_1 \wedge A_2 \wedge A_3 \wedge A_4 \wedge (C = k)$. $N_d(k) = N_T - (M_d(k) - Y_d(k))$ is the number of useful instances for the calculation of the measurement. *support* and *confidence* of the rule $A_1 \wedge A_2 \wedge A_3 \wedge A_4 \rightarrow (C = k)$ become

$$support = \frac{Y_d(k)}{\sum_{i=0}^{K-1} N_d(i)}, \quad confidence = \frac{Y_d(k)}{\sum_{i=0}^{K-1} Y_d(i)}.$$

Table 2 shows an example of the measurements of the CARs generated by the node connections in Fig. 1.

In every generation of GNP, the examinations are done from $1 \in ID$ and P_1 node. When all the instances are examined, measurements of candidate rules of every processing nodes are calculated. Measurements of the rules are calculated and the interestingness of the rules are judged by given conditions. When a candidate rule is extracted by GNP, whether the same rule is in the rule pool or not is checked. The extracted important rules are stored in the pool all together through generations.

3.4 Genetic Operations and Fitness Function

The rules produced by changing the connections of GNP or the rules changing some attributes could be candidates of important rules. We can obtain these rules effectively by GNP genetic operations, because mutation and crossover change the connections or contents of the nodes. Following three kinds of genetic operators described in [8] are used; crossover, mutation-1 (changes the connection of nodes) and mutation-2 (changes the function of judgment nodes).

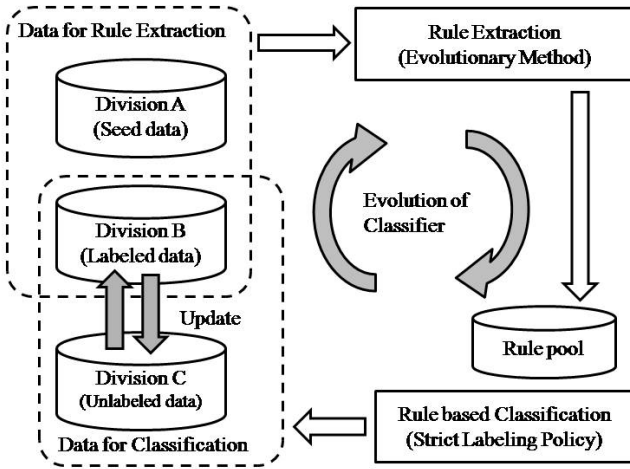


Fig. 3. Flow of the proposed evolving classifier

Fitness of GNP individual is defined considering potential of new candidate rule representation. Following fitness was used for the experiments in Section 5.

$$F_{class} = \sum_{r \in R} \{ \chi^2(r) + 10(n_X(r) - 1) + \alpha_{new}(r) \}.$$

where, R : set of suffixes of extracted rules satisfying (1), (2) and (3) in a GNP individual,

$\chi^2(r)$: χ^2 value of rule r ,

$n_X(r)$: the number of attributes in the antecedent of rule r ,

$\alpha_{new}(r)$: additional constant defined by

$$\alpha_{new}(r) = \begin{cases} \alpha_{new} & (\text{rule } r \text{ is new}) \\ 0 & (\text{otherwise}). \end{cases}$$

$\chi^2(r)$, $n_X(r)$ and $\alpha_{new}(r)$ are concerned with the importance, complexity and novelty of rule r , respectively. $n_X(r)$ is also concerned with the existence of good judgment node connections in the GNP individual.

4 Evolving Classifier Using Labeled Instances

A method of building an evolving classifier is proposed using GNP based rule extraction method. The aim of the evolution of it is to gather useful information for classification and improve the ability of classifier. In the GNP based method, rules satisfying the given conditions are stored in a rule pool through GNP generations. The rule extraction can be quit anytime when the number

Table 3. Averaged cover rate of rule extraction (100 trials)

	Missing rate (%)				
	0	5	10	20	33
Number of expected rules for extraction (Interesting rules *) (Number of long rules: $n_X(r) \geq 7$)	6817 (1310) (1146)	4035 (788) (297)	2193 (429) (6)	822 (188) (0)	310 (87) (0)
Cover rate at 50 generations (%) (Interesting rules *)	77.4 (97.9)	87.3 (99.3)	93.9 (99.8)	98.2 (99.9)	99.7 (100.0)
Cover rate at 200 generations (%) (Interesting rules *)	95.8 (99.9)	98.5 (99.9)	99.4 (99.9)	99.9 (100.0)	100.0 (100.0)

(*):Results for the rules satisfying additional conditions within the rule pool.

of rules given by users in advance for building a classifier are obtained. Applying extracted rules at the moment for classification, newly labeled instances can be obtained. If these just labeled data are joined into the training data, then extended training data can be constructed. We can repeat this operation and evolve the classifier using acquired data which labeled by itself. Fig. 3 shows the flow of the proposed evolving classifier. A cycle of rule extraction and classification is repeated until given finish condition. One cycle for classification is defined as *round* for a concept of upper layer of evolutionary process.

In the proposed method, data are divided into three categories as follows:

- Division A (Div-A): Set of the seed instances. This part works as training data for the first building a classifier. All the instances are labeled in advance and used for training data through evolutionary process. Re-classification accuracy for this division can monitor the performance of the classifier.
- Division B (Div-B): Set of labeled instances by the evolving classifier. This division is empty at the initial round.
- Division C (Div-C): Set of unlabeled instances.

All the instances in Div-A and -B are used for GNP based rule extraction for each class label. If the number of extracted rules are enough for given condition or spent a given number of GNP generation, then stop the rule mining and build a classifier. After the classification, we empty the rule pool in order to extract rules for the next classification.

The instances in Div-B and -C are labeled based on the method described in subsection 2.2. In order to obtain good labeling with confidence, $\max\{score(k)\} \geq score_{min}$ is used, where, $score_{min}$ is the threshold value given by users. In the case of $\max\{score(k)\} < score_{min}$, we did not label the instance. When an instance in Div-B is not labeled, then the instance moves to Div-C. At the end of the evolution, the gathered data in Div-B bring us discovered information. Instances left in Div-C can include candidates of unknown or abnormal cases.

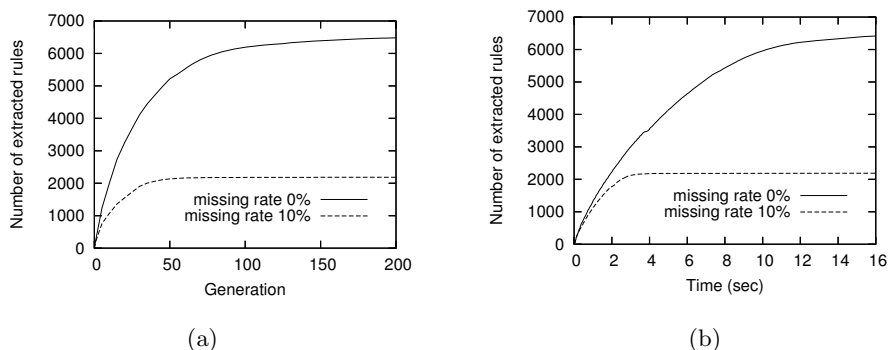


Fig. 4. Number of extracted association rules in the pool (cleveland ($C=1$))

5 Experimental Results

5.1 Classification for Incomplete Database

cleveland and *breast-w* data set from UCI ML Repository¹ were used for the evaluation. All the continuous attribute values are transformed to a set of attributes, whose value is 1 or 0 using the Entropy method. The transformed *cleveland* and *breast-w* dataset are complete and consist of 30 attributes, 297 instances and 18 attributes, 683 instances, respectively. Both data sets have 2 classes for classification, respectively. The artificial incomplete data sets were generated using random missing values with missing rates 5%, 10%, 20% and 33%. 30 data set for each missing rate were generated.

CARs are extracted for each class using each specific GNP. $supp_{min} = 0.05$ in (1), $\chi_{min}^2 = 6.63$ in (2), (3) and $1 \leq n_X(r) \leq 8$ were used as definition for the interestingness. The population size of GNP is 120. The number of processing nodes and judgment nodes in each GNP individual are 10 and 75, respectively. $P_c = 1/5$, $P_{m1} = 1/3$, $P_{m2} = 1/5$ and $\alpha_{new} = 150$ was used. The condition of termination is 200 generations. All algorithms were coded in C. Experiments were done on a 2.80 GHz Intel(R) i7 860 with 4 GB RAM.

Table 3 shows that the averaged cover rate of rule extraction using *cleveland* data from each missing rate. Cover rate (%) is defined as

$$\frac{\text{Number of extracted rules}}{\text{Number of expected rules for extraction}} * 100.$$

Rule extraction was done for ($C = 1$) and 0% means the results using complete data. As the GNP based method cannot guarantee the extraction of all rules by the nature of evolutionary discovering policy, cover rate is an important factor of performance study. The number of expected rules is defined as the number of identified rules from independent 100 rule extractions for the focused database.

¹ <http://www.ics.uci.edu/mllearn/MLRepository.html>

Table 4. Classification accuracy versus missing rate (averaged over 30 data sets)

(a) Cleveland

Missing rate (%)	Treat as missing (Number of rules)			Treat as absence (Number of rules)		
	100	250	1000	100	250	1000
0	83.33	83.95	83.75	—	—	—
5	83.15	83.34	83.51	83.19	83.21	83.54
10	82.85	83.15	83.05	82.82	83.13	83.02
20	82.09	82.41	82.13	81.83	81.96	81.98
33	80.88	81.19	81.10	80.02	80.02	80.20

(b) Breast-w

Missing rate (%)	Treat as missing (Number of rules)			Treat as absence (Number of rules)		
	100	250	1000	100	250	1000
0	97.43	97.71	97.80	—	—	—
5	97.37	97.49	97.49	97.39	97.50	97.51
10	97.21	97.30	97.28	97.19	97.31	97.27
20	96.95	96.94	96.92	96.89	96.97	96.98
33	96.35	96.45	96.45	96.32	96.24	96.31

Table 5. Classification accuracy versus missing rate (cleveland, averaged over 30 trials)

Test data missing rate (%)	Training data missing rate (%)				
	0	5	10	20	33
0	86.3	86.2	86.4	86.1	86.1
5	86.1	85.9	86.1	85.4	85.2
10	85.3	85.3	85.0	85.2	84.8
20	85.4	85.2	85.2	85.0	84.3
33	83.6	83.8	84.2	84.2	82.9

It is found that almost the expected rules for this experiment can be extracted at 200 generations. Interesting rules in the Table 3 is defined as the rules satisfying following additional conditions within the rule pool: $support(X \rightarrow (C=1)) \geq 0.1$ and $\chi^2(X \rightarrow (C=1)) \geq 10.0$. Almost the important rules are extracted at 50 generations. The rules with high measurement values of interestingness tend to be extracted easily. The number of extracted rules depends on the missing rate. As the missing rate increases, the number of long rules like satisfying $n_X(r) \geq 7$ decreases. Fig. 4(a) shows examples of the number of rules obtained in the pool under no missing data and 10% missing data, respectively. Fig. 4(b) shows the number of extracted rules versus run-time in the same experiment as Fig. 4(a). It shows that the number of extracted rules within 2 seconds are almost the same in the cases of no missing data and 10% missing data, therefore, the proposed method does not require long run-time even when there are missing data.

	A1-A25	A26-A50	A51-A75	A76-A100	Class (#data)
Known (Division A)			Missing values		C=1 (30) C=0 (30)
	Missing values			Missing values	C=1 (30) C=0 (30) Noise (30)
Unknown (Division C)	Missing values				C=1 (30) C=0 (30) Noise (30)
			Missing values		Noise (30)

Fig. 5. Data setting for evaluation of the evolving classifier

The performances of classification was evaluated using Leave-One-Out validation evaluation. Each extracted rule in the pool has an inherent numeric order by its appearance. The classification performance was examined using 100, 250 and 1000 rules for each class by the order of appearance in the rule pool. $score_{min} = 0$ was used for the classification. The classification results are shown in Table 4. The column 'Treat as missing' shows the results by the method described in Section 3. They showed that the accuracy of classification is favorable even if some instances include missing values. The column 'Treat as absence' shows that the classification results in the case that the missing values are regarded as '0' instead of 'm'. The aim of this experiment is to simulate item based classification without missing value information. The difference between both methods in this experiment is not so large. This can be caused by multiple matched rule policy of the method.

The classification accuracy for combination of different missing rate between training and testing data was examined using *cleveland*. 10% of the instances were selected randomly for test data and remaining 90% were used for training. 30 combinations of training and test data were generated. Table 5 shows the classification results. It is found that the accuracy depends on the missing rate of testing data. This means that even if the missing data exist in the training data, we can build a stable classifier.

5.2 Evaluation of Evolving Classifier

A dense database with 100 attributes was used for the evaluation. The original data is The Mapping 500K HapMap Genotype Data Set (Affymetrix) [2]. This database contains Single Nucleotide Polymorphism (SNP) information of 270 people. The same data named SNP_{com} described in [8] was used. SNP_{com} has 100 attributes and 270 instances with no missing values. Original data has 4 class labels: CEU (90 instances), YRI (90 instances), JPT (45 instances) and CHB (45 instances). We defined as $C = 1$ in the case of CEU, $C = 0$ in the case

² http://www.affymetrix.com/support/technical/sample_data/500k_hapmap_genotype_data.affx

Table 6. Number of classified instances by evolving classifier. (No noise data)

Generation of GNP	Division (Class)	# classified instances	Accuracy (%)
0 (Round 0)	A (C=1)	30	—
	(C=0)	30	—
	B (C=1)	—	—
14 (Round 1)	(C=0)	—	—
	A (C=1)	32	93.8
	(C=0)	26	100.0
	B (C=1)	30	96.7
22 (Round 2)	(C=0)	22	95.5
	A (C=1)	33	90.9
	(C=0)	25	100.0
	B (C=1)	60	100.0
	(C=0)	58	100.0
32 (Round 3)	A (C=1)	32	93.8
	(C=0)	27	100
	B (C=1)	59	100
	(C=0)	57	100
...			
137 (Round 13)	A (C=1)	31	96.8
	(C=0)	28	100.0
	B (C=1)	60	100.0
...	(C=0)	59	100.0
199 (Round 19)	A (C=1)	32	93.8
	(C=0)	27	100.0
	B (C=1)	61	98.4
	(C=0)	58	100.0

of YRI. JPT and CHB were used as noise. Instances were divided for the initial data randomly and modified using artificial missing values as follows (See Fig.5). Class labels of instances in Div-C are used for the evaluation.

- Div-A: 30 instances for $C=1$ and 30 instances for $C=0$ (attribute values of $A_{51}-A_{100}$ are missed)
- Div-C: 30 instances for $C=1$, 30 instances for $C=0$ and 30 instances for Noise (attribute values of A_1-A_{25} and $A_{76}-A_{100}$ are missed)
- Div-C: 30 instances for $C=1$, 30 instances for $C=0$ and 30 instances for Noise (attribute values of A_1-A_{50} are missed)
- Div-C: 30 instances for Noise (attribute values of $A_{51}-A_{100}$ are missed)

The same parameter settings for the GNP described in 5.1 was used except use of 100 judgment nodes in each GNP individual. $supp_{min}=0.03$ for (1), $\chi_{min}^2=3.84$ for (2) and $2 \leq n_X(r) \leq 6$ was used. Instead of (3), $confidence(X \rightarrow (C=k)) \geq 0.7$ was used. When 200 rules for each class label were extracted, then the classification was done. $score_{min}=q$ ($q=0.05, 0.1$ and 0.3) were used for the

Table 7. Averaged number of classified instances and accuracy by evolving classifier

Division (Class)	$score_{min} = 0.05$		$score_{min} = 0.1$		$score_{min} = 0.3$	
	# classified instances	Accuracy (%)	# classified instances	Accuracy (%)	# classified instances	Accuracy (%)
A (C=1)	34.0	88.2	33.7	89.1	27.4	97.6
(C=0)	25.8	99.9	24.3	99.9	14.6	100.0
B (C=1)	108.3	55.6	101.4	59.2	48.7	93.1
(C=0)	99.4	60.5	90.5	64.1	42.0	85.4

class label prediction. In the case of $\max\{score(k)\} < q$, the instance was not labeled. If the unlabeled instance was in Div-B, then it moved to Div-C. After the classification, the rule pool was emptied in order to extract new rules for the next classification. In addition, judgment node functions of GNP individuals were initialized after each classification. The number of GNP generation was counted continuously. Accuracy (%) for each *round* is defined as

$$\frac{\text{Number of correctly classified instances}}{\text{Number of classified instances}} * 100.$$

First of all, the performance in the case of no noise data, that is, used 180 instances of $C=1$ and $C=0$ was tested. Table 6 shows an example of the number of classified instances, that is, the number of instances gathered into Division B, and the classification accuracy. Instances having missing values of A_1-A_{25} and $A_{76}-A_{100}$ were labeled at generation 14 and the number of instances in Division B increased gradually. Eventually, almost the instances having missing values of A_1-A_{50} were labeled. It was found that the prediction accuracy of the instances in Division B was also improved through generations. The accuracy of re-prediction for Division A did not always good compare to Division B. One of the reason of this can be that the proposed method discovered the more useful information for the classification from the data in Division C. The results showed that evolving classifier using GNP based method has a potential of information gathering from the unlabeled instances. Calculation time of the method is very short, because run time of the GNP based rule mining depends on the number of nodes in GNP individual and the number of extracted rules in the rule pool. Generally, the number of attributes in the data set is independent of calculation time of GNP based method.

Table 7 shows the averaged number of the classified instances and accuracy of the classification at 200 GNP generations over 30 trials. Using a strict condition like $score_{min} = 0.3$ avoided the wrong labeling, however, the number of labeled instances in Div-B decreased. How to define this parameter depends on a problem and user's needs. The results showed that the proposed evolving classifier has a potential of information gathering from the unlabeled data.

6 Conclusions

A classification method for incomplete databases using GNP has been demonstrated. The performance of the rule extraction and classification were estimated using artificial incomplete data. The results showed that the accuracy of classification is favorable even if some instances include missing values. In addition, an evolving classification method was proposed using the above rule extraction method. The method evolves the classifier using the labeled instances by itself in the previous classification. The results of experiment using incomplete data showed that the proposed method has a potential of gathering information. We are studying applications of the proposed method to information processing in the medical science field.

References

1. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: Proc. of the 20th VLDB Conf., pp. 487–499 (1994)
2. Liu, B., Hsu, W., Ma, Y.: Integrating Classification and Association Rule Mining. In: Proc. of the ACM Int'l Conf. on Knowledge Discovery and Data Mining, pp. 80–86 (1998)
3. Li, W., Han, J., Pei, J.: CMAR: Accurate and efficient classification based on multiple class-association rules. In: Proc. of the 2001 IEEE Int'l Conf. on Data Mining, pp. 369–376 (2001)
4. Yin, X., Han, J.: CPAR: Classification based on predictive association rules. In: Proc. of the 2003 SIAM Int'l Conf. on Data Mining, SDM 2003 (2003)
5. Baralis, E., Chiusano, S., Garza, P.: A Lazy Approach to Associative Classification. *IEEE Trans. on Knowledge and Data Eng.* 20, 156–171 (2008)
6. Grzymala-Busse, J.W., Grzymala-Busse, W.J.: Handling Missing Attribute Values Data Mining and Knowledge Discovery Handbook. In: Maimon, O., Rockach, L. (eds.) Springer Science, 2nd edn., pp. 33–51 (2010)
7. Saar-Tsechansky, M., Provost, F.: Handling Missing Values when Applying Classification Models. *Journal of Machine Learning Research* 8, 1625–1657 (2007)
8. Shimada, K., Hirasawa, K.: A Method of Association Rule Analysis for Incomplete Database Using Genetic Network Programming. In: Proc. of the Genetic and Evolutionary Computation Conference 2010 (GECCO 2010), pp. 1115–1122 (2010)
9. Shimada, K., Hirasawa, K., Hu, J.: Association Rule Mining with Chi-Squared Test Using Alternate Genetic Network Programming. In: Perner, P. (ed.) *ICDM 2006*. LNCS (LNAI), vol. 4065, pp. 202–216. Springer, Heidelberg (2006)
10. Shimada, K., Hirasawa, K., Hu, J.: Class Association Rule Mining with Chi-Squared Test Using Genetic Network Programming. In: Proc. of the IEEE Conf. on Systems, Man, and Cybernetics, pp. 5338–5344 (2006)
11. Mabou, S., Chen, C., Lu, N., Shimada, K., Hirasawa, K.: An Intrusion-Detection Model Based on Fuzzy Class-Association-Rule Mining Using Genetic Network Programming. *IEEE Trans. on Systems, Man, and Cybernetics - Part C* - 41, 130–139 (2011)

Improving Classifier Performance by Knowledge-Driven Data Preparation

Laura Welcker¹, Stephan Koch², and Frank Dellmann¹

¹ Münster University of Applied Sciences, Münster, Germany
{laurawelcker,dellmann}@fh-muenster.de

² BBDO Proximity GmbH, Hamburg, Germany
Stephan.Koch@bbdoproximity.de

Abstract. Classification is a widely used technique in data mining. Thereby achieving a reasonable classifier performance is an increasingly important goal. This paper aims to empirically show how classifier performance can be improved by knowledge-driven data preparation using business, data and methodological know-how. To point out the variety of knowledge-driven approaches, we firstly introduce an advanced framework that breaks down the data preparation phase to four hierarchy levels within the CRISP-DM process model. The first 3 levels reflect methodological knowledge; the last level clarifies the use of business and data know-how. Furthermore, we present insights from a case study to show the effect of variable derivation as a subtask of data preparation. The impact of 9 derivation approaches and 4 combinations of them on classifier performance is assessed on a real world dataset using decision trees and gains charts as performance measure. The results indicate that our approach improves the classifier performance.

Keywords: classification, framework for data preparation, knowledge-driven data preparation, decision trees.

1 Introduction

Classification is one of the most important and widely used data mining techniques, especially in the area of marketing and Customer Relationship Management (CRM) [1,2]. Due to the widespread use of classification applications in today's highly competitive sectors, continuous improvement of its predictive power has gained importance. Approaches for improvement of classifier performance can be categorized according to the data mining process and its distinct steps. Common process models describing the data mining process are the KDD (Knowledge Discovery in Databases) process model [3], SEMMA [4], Reinartz's framework [5] and CRISP-DM [6]. All process models include a step dealing with data preparation. This step is often referred to as the most time consuming but also the most important part of the data mining process [7,8]. To outline the importance we clarify the risks that occur when accurate data preparation is lacking. First, the data might not meet the requirements of the

algorithms in use. Second, poor or no data preparation is likely to lead to an incomplete and inaccurate data representation space, which is spanned by variables and realizations used in the modeling step. Both risks may dramatically affect the classifier performance and can lead to poor prediction accuracy or even in wrong predictive models. From the authors' experience, the potential of improved data preparation often remains unused by practitioners. They fail to transform their understanding of business and data into a properly prepared representation space. Reasons for this could be a lack of time and/or methodological know-how. That is why many researchers and especially software vendors try to enforce the automation of data preparation that guarantees an ease of use and time savings. However, automation is where knowledge-driven data preparation becomes impossible because the selection and evaluation of preparation instruments in automated data preparation strongly relies on mathematical and statistical methods, instead of relying on business, data and methodological understanding. We support the assumption that the inclusion of knowledge-driven data preparation has a positive impact on classifier performance contrasted to the exclusion of it.

Many authors state that the role of the human within the data mining process is essential [9,10,11]. This approach is often described as domain knowledge. "Domain knowledge consists of information about the data that is already available either through some other discovery process or from a domain expert" [9, p. 37]. Furthermore, [9] claim that domain knowledge can affect the discovery process within the data mining system in two ways. First, it can make patterns more visible by generalizing the variable realizations, and second, it can constrain the representation space as well as the rule space. [10] analyses the question how domain knowledge can be used to evaluate and interpret classification models. The study of [11] concentrates on the use of domain knowledge in all phases of the data mining process. In this paper the knowledge-driven approach can be differentiated from domain knowledge as it is defined as business, data and methodological know-how. Moreover, the knowledge-driven approach focuses on its integration only within the data preparation phase. A study, which deals with the same topic as this paper comes from [12]. The authors compare the performance of classification models with and without domain knowledge. But they express their domain knowledge only by one categorical variable derived from an expert. In this study, we show the multitude and power of knowledge-driven approaches by applying more than one derivation approach on a large number of variables.

The paper's objective is to show that the use of knowledge-driven data preparation leads to higher classifier performance in comparison to standard preparation (see section 3). To reach this objective, the paper reports on a case study applying knowledge-driven data preparation. In this study a classifier is built on a standard dataset, which was extended through knowledge-driven derivation approaches. The resulting classifier is compared to a reference classifier, which was built only on the standard dataset (without the extension). The findings and results gained from the study are major contributions of this paper.

Furthermore, a variety of data preparation tasks are structured within a framework in order to present a compilation of methods as well as a comprehensive and procedural guideline. Most of the preparation tasks has already been mentioned in literature [8,13,14,15], but a framework, which lists and structures all of the identified approaches, does not exist to the authors' knowledge. Therefore, the proposed framework for knowledge-driven data preparation can be considered as a further contribution of this paper.

The paper is divided into four sections. After the introduction, section 2 introduces the advanced framework based on data preparation and describes the step of variable derivation in further detail, since it is the basic concept of the study. Section 3 reports on the case study by describing the experimental setup and the results. Section 4 provides a conclusion by assessing the results and showing further research fields as well as limitations.

2 Introducing an Advanced Framework Focused on Data Preparation

2.1 Advanced Framework for Data Preparation

The advanced framework for data preparation is based on the first two levels of the CRISP-DM methodology. We focus on CRISP-DM, since it is considered as the most complete [16,17] and broadly adopted data mining process model [7]. It provides a systematic overview of the life cycle of a data mining project and consists of six major phases. Even if the original aim of CRISP-DM was already to make data mining projects such as classification projects "more efficient, better organized, more reproducible, more manageable and more likely to yield business success" [18], the necessity for more specific and detailed framework was also proclaimed by [6, p. 11] in the context of "mapping for the future". However, this update of the CRISP-DM methodology (named CRISP-DM 2.0) was only initialized [18], but has not revealed any official results to this date.

Referring to the idea of CRISP-DM 2.0 we designed our advanced framework focused on data preparation because of its high impact on the quality of the classifier performance. This impact has been experienced by us in numerous practical projects and is also stated in literature [19]. In computer science for instance, the impact of low data quality on the output quality has been discussed under the "Garbage in garbage out"-principle [20,21].

To implement the advanced framework we broke down the second level of the CRISP-DM process model, the generic tasks into three additional levels: subtasks, steps, and realization options (see Figure 1). With these additional levels the user gets a deeper insight into the data preparation opportunities and can more easily decide which approach would reasonably improve the individual classification issue. The five generic tasks for data preparation represents an adequate baseline for our framework because of its completeness and discriminatory power, which is beneficial to the practitioners. The five generic tasks are distinct from each other. They are sorted in chronological order as the user will

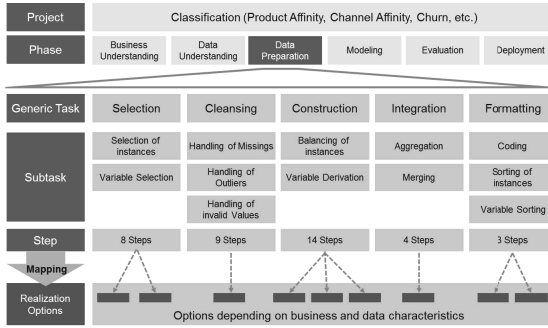


Fig. 1. Advanced Framework for Data Preparation [Source: Authors' own construct]

need them in a real project. Nevertheless, having feedback loops between the generic tasks is always required. Each generic task is divided into a complete set of subtasks as detailed below.

1. **Selection:** The generic task selection includes all subtasks to select valid and relevant instances and variables. In terms of instances, the selection contains among others a form of complexity reduction by the step of splitting (in case of very heterogeneous instances, e.g. private and business customers). Selection of variables in turn comprises for example the steps of time reference and elimination of multicollinearity.
2. **Cleansing:** This generic task contains all steps to replace, keep, bin and, if necessary, eliminate conspicuous data such as outliers, missing values and invalid instance values.
3. **Construction:** Construction is divided into two subtasks: balancing of instances and derivation of new variables. Balancing of instances can be performed by oversampling and undersampling. Derivation of new variables is categorized into univariate (e.g. normalization), multivariate (e.g. factor analysis) and hybrid approaches (e.g. segmentation with classified variables). Further information about the variable derivation gives subsection 2.2.
4. **Integration:** Integration deals with the aggregation (multiple rows to single row) and disaggregation of instances (single row to multiple rows) as well as merging of datasets.
5. **Formatting:** The last generic task contains the steps to adjust value coding and to sort instances and variables in accordance with the software and algorithms requirements.

The subtasks consist of different steps, which are only counted and not further described within this paper. Up to the level of steps the framework has a general validity. At level 4, a specialized mapping has to be conducted because the realization options depend on business and data characteristics and cannot be defined on a generic basis.

We have to consider that our framework has certain limitations. Although we derived numerous realization options for all steps, our framework cannot claim

to be exhaustive. Applying this framework the practitioner only has to decide which subtasks, steps and realization options are relevant in the specific context. Compared to existing process models this deeper hierarchical structure facilitates the analyst's job. He can more easily transform his knowledge about business, data and methods into a relevant dataset.

2.2 Variable Derivation

In this paper we examine in depth the step of deriving new variables because we are of the opinion that a knowledge-driven variation of the representation space has a greater potential to improve classifier performance than e.g. the reduction of it, caused by selection or cleansing. The relevance of the other generic tasks is due to further research. Moreover, the subtask of variable derivation is closely linked to the business and data understanding of the user. [14] state that the most effective derived variables are those, that express additional information (beyond the database), such as a description of some underlying customer behavior. For the creation of useful derivations it is important to use background or domain knowledge and not to randomly combine a large number of variables. That is the reason why automated tools are not well-suited to produce valuable results by creating derived variables.

In the area of Machine Learning, derivation of new variables is referred to constructive induction, which was introduced by [22]. Subsequently, [23] presented three different constructive induction strategies:

- **Hypothesis-driven:** Changes are based on hypothesis, generated in each data analysis-iteration and the discovery of patterns.
- **Data-driven:** Data characteristics are used to generate new data representation spaces.
- **Knowledge-driven:** Expert domain knowledge is applied.

The combination of two or more of these strategies is denominated as multi-strategy constructive induction [23]. Our approach of variable derivation can be classified as multi-strategy constructive induction as we combine data-driven and knowledge-driven strategies.

A considerable amount of literature has been published on variable derivation [24,25,26]. In order to develop a complete framework we collected various approaches and put them into a separate framework shown in Figure 2. The subtask of variable derivation consists of eleven distinct process steps. Corresponding to the number of variables involved in a step, they are categorized in univariate and multivariate approaches. Approaches combining multiple options are categorized as hybrid approaches. All approaches are broken down into various realization options, which are represented on the last level of our framework.

- **Univariate Approaches:** The first step called *development* includes all derivations representing the development of a variable in time. Ratios as well as absolute values could serve this purpose. On the next level of our framework we split these options up by scale and operator. *Normalization* is

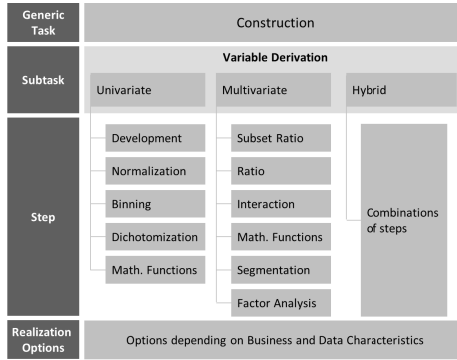


Fig. 2. Breakdown of the subtask variable derivation [Source: Authors’ own construct]

the next step and deals with all kinds of scale harmonization. The most commonly used form of normalization is the input standardized to zero mean and unit variance. In our framework the *binning* step comprises all groupings of values irrespective of the variable’s scale. Realization options break it down to options for each scale. The step of *dichotomization* might be essential in data mining projects even without knowledge-driven acting. Due to algorithms’ requirements, dichotomization is often used to transform categorical variables into binary variables, where each derived variable represents one value of the former variable. From the business point of view dichotomization can be used to emphasize important variables and values. Derivations realized by applying a *mathematical function* to an input variable are consolidated in the last step of univariate approaches. Logarithmic transformation is a well known realization option within this step.

- **Multivariate Approaches:** *Subset ratios* are calculated within the first multivariate step in order to reflect the structure of a given entity subset. Consequently, the numerator is always a subset of the denominator. Ratios between hierarchy-independent variables are separately categorized in the step named *ratio*. Whereas the above mentioned multivariate approaches focus on the metric scale, *interactions* allow the practitioner to combine categorical variables. New values are derived from the matrix of the original value pairs. Combinations of two or more variables by means of other *mathematical functions* are summarized in a further step. *Segmentation* and *factor analysis* are quite complex approaches to derive new variables as they represent small data mining projects on their own. However, these two steps can be useful especially when dealing with huge numbers of variables or observing multicollinearity. Depending on the input data there are different realization options for both analyses. The study of [27] shows that a knowledge-driven derivation of segmentation variables can outperform a random selection of variables with regard to performance gain.
- **Hybrid Approaches:** In the category of hybrid approaches, deriving new variables by using more than one of the steps mentioned above is considered

as another powerful way to improve predictive accuracy by widening the representation space. This last step enables the practitioner to transfer more complex business know-how into the data.

Finally, Table 1 summarizes all steps by giving a short description and an example. The examples mostly refer to the financial sector because they are adopted from our case study, which is described in the next section.

Table 1. Derivation Approaches [Source: Authors’ own construct]

Category	Step	Description	Example	
Uni-variate	Development	Development of a variable over at least two points in time, expressed e.g. as difference or ratio	Ratio from asset under management at t0 to assets under management at t-6	
	Normalization	Harmonization of a variable's scale by normalization techniques	Standardization of all product assets (zero mean and unit variance)	
	Binning	Reduction of values by grouping	Grouping of postal codes	
	Dichotomization	Deriving a binary variable from one value of multiple variable values	Flagging customers with zero balances on deposit accounts as passive	
	Mathematical Function	Applying a mathematical function to a variable	Logarithm of distance to next branch	
Multi-variate	Subset Ratio	Relation between two variables, where the numerator is a subset of the denominator	Ratio between credit volume and total assets under control	
	Ratio	Relation between two variables that do not have a hierarchical connection	Ratio between transaction volumes and age	
	Interaction	Deriving new values from the matrix-combination of values from categorical variables	Combination of region and income class	
	Mathematical Function	Combining variables by further mathematical functions	Income of a customer divided by the median income of all customers; the amount of products within a certain division	
	Factor Analysis	Applying factor analysis to derive new variables (factors)	Factors representing the use of credit cards by sectors	
Hybrid	Examples for Combinations of Steps	Segmentation	Applying segmentation algorithms in order to receive the segment labels as values for a variable	Client segments derived from profiling of transactional data
		Development of multivariate mathematical functions	Development of the amount of certain products over time	
		Development of subset ratios	Development of credit volume divided by total assets under control over time	
		Dichotomization of binning variables	To bin the values of assets under control into 4 groups and flag these groups afterwards	
		Ratio of two mathematical functions	Logarithm of income divided by the logarithm of age	

3 Case Study on Variable Derivation

After presenting our advanced framework for data preparation in general and the subtask of variable derivation in more detail, this section deals with the practical application of these approaches.

3.1 Experimental Setup

Research Domain and Data. The conducted study was realized with customer data from the German financial retail sector. The raw dataset comprised about 1.7 million instances and 541 potential predictor variables. The underlying direct marketing issue was defined as building an affinity score for closed funds in order to support the selection of relevant target groups for direct mail. In this context buying a closed fund as a reaction of a direct mailing was designed as binary target variable with "1" for customer reaction and "0" for non-reaction. In order to keep data free of any causal reference between product sale and predictor values we set up a time-dependent selection of potential predictor variables based on the individual acquisition date of each customer.

Research Design. Figure 3 displays the test setting, which was defined to evaluate the impact of knowledge-driven derivation on classifier performance. The methodology is basically marked by a comparison of two classifiers that only differ in the composition of their input data. Based on the raw dataset (541 variables), measures of standard data preparation are conducted. Consequently, the raw dataset gets reduced by x variables. The dataset created by this means (541 - x variables) is the basis for the reference model, the REF classifier. For the other classifier the dataset is extended through knowledge-driven derivation by y variables, so that the classifier ALL has 541 - x + y potential input variables. The two classifiers are both built the same way by the same classification technique with the same parameters. The only difference between the two are the y derived variables, so that every difference in performance can clearly be assigned to the impact of variable derivation.

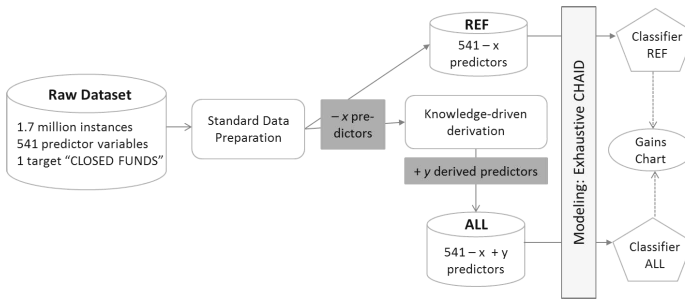


Fig. 3. Test Setting [Source: Authors' own construct]

Classification Technique. In this study, we used PASW Modeler 14 as data mining software to run the classification. Classification models within this project were created by a decision tree algorithm. According to [112] decision trees and regression analysis are the most commonly used data mining techniques. Furthermore, research results do not show a systematical outperformance of other techniques [28,29]. Due to their ease of use and the ability of some tree algorithm to handle missing values as separate categories, we preferred decision trees to regression analysis. In many cases the separate handling represents best the real situation. Missing values for instance were created when a development of the variable "product volume" could not be calculated due to a very recent acquisition. For using decision trees the applied software provides four different algorithms: CHAID, C5.0, QUEST and CART. Due to data characteristics we preferred tree algorithms that theoretically allow multiway splits like CHAID and C5.0. We expected them to perform better on the given data than CART or QUEST with regard to quality and clearness. The CHAID algorithm was finally chosen for this project because it tends to produce shallower trees than C5.0 with a higher level of multiway splits.

The CHAID algorithm is originally proposed by [30] and its application is popular in marketing. It uses an attribute selection measure that is based on the statistical Chi-squared test for independence [15]. The software also offers "Exhaustive CHAID", which is a refinement of CHAID proposed by [31]. Both algorithms CHAID and Exhaustive CHAID allow multiple splits of a node rather than binary or c-way splits (where c is the original number of categories of the predictor variable). The difference lies in the merging step, where Exhaustive CHAID uses an exhaustive search procedure to merge any similar pair until only a single pair remains [31].

The Exhaustive CHAID algorithm was used with Bonferroni correction to build trees of five levels. Chi-square was calculated based on Pearson and the split of merged nodes was permitted. Further parameters were kept as default in PASW Modeler 14.

Performance Measure. The performance measure in this project was a gains chart. The gains chart plots the "Gains (%)" on the y-axis and percentiles on the x-axis. Gains are defined as the proportion of hits in each increment relative to the total number of hits in the classifier. Gains charts effectively illustrate how widely the practitioner needs to cast the net to capture a given percentage of all of the hits in the tree [14,17]. This performance measure has been chosen because it is really suitable to compare two classifiers in a marketing context. Moreover, gains charts can handle imbalance class problems better than accuracy or classification error [14].

Standard Data Preparation. In course of the standard data preparation the raw dataset has been divided into a training (80%) and a test (20%) set. In consequence 345,417 instances were kept as test partition. Due to a skewed distribution of the target classes (only 0.12% for the 1-class), a balancing of instances was necessary. Several ratios of downsizing were tested to examine the best weighting of positive instances (1-class) in the training set. A ratio of 10% of positive instances clearly outperformed the other tested ratios (5% and 15%) and was subsequently applied. The balancing led to a distribution of 15,147 negative instances and 1,683 positive instances in the training set.

As a first step the given raw dataset was prepared by the adjustment of scale levels and by filtering variables. Variables were filtered out if they meet one of the following filter criteria:

1. Quality criteria: Dominance of missings or single values/categories, multi-collinearity etc.
2. Target-specific criteria: Contextual irrelevance for the target (e.g. the name of the customer)

The raw dataset comprises with 541 variables all available company information, but the classification task is only focused on closed funds. Therefore, most of the variables were filtered out due to contextual irrelevance for the classification problem. To decide if a variable is relevant or irrelevant for the underlying

task the importance of domain knowledge and the integration of a domain expert into the selection process were again emphasized. No variable had to be excluded due to dominating the tree construction in the first branches. In course of the standard data preparation 270 predictor variables were filtered out (with reference to Figure 3 is $x = 270$).

Variable Derivation. In course of knowledge-driven derivation, 50 variables were directly transformed by normalization and 201 derived variables were added to the dataset (thus named ALL) by knowledge-driven derivation (with reference to Figure 3 is $y = 201$). Table 2 lists all approaches that were conducted within this study. We only accepted derivations with contextual relevance. Due to restrictions in time and data limitations the implementation of further derivation approaches, such as factor analysis and segmentation was excluded from this study. Within the step *development* 41 new variables were derived as trends over one, three and six months. Changes on longer terms could not be calculated due to structural changes in the database. Besides, we did not expect long term changes to have extra predictive power based on our business experience. The calculated trends included assets under management, account balances, transaction volumes, ratings and other variables. *Normalization* was conducted for 50 metric variables. The standardized variables were the only derivations, which have directly replaced the original variables as we did not want to emphasize their information disproportionally. *Binning* was used to make the information of variables with many categories more accessible to the algorithm. We grouped for example postal codes in order to present this regional information on a higher level. As passive customers often show very special behaviors, we flagged those with zero balances on deposit accounts within the *dichotomization* step. The univariate application of a *mathematical function* was only used for one variable (distance to next branch).

The multivariate approach of *subset ratios* led to 33 new variables. Some of them were built to represent the overall asset structure, for example ratio of credit volume to total assets under control, ratio of deposit volume to total assets under control, and ratio of daily allowance to total assets under control. Further variables were calculated to reproduce the structure of a certain product, for example the customer account: Ratios of investments by region, ratios of investments by paper types, and ratio of investments by investment sector. The

Table 2. Applied Derivation Approaches [Source: Authors' own construct]

Univariate Steps	Nb. of derived variables	Multivariate Steps	Nb. of derived variables	Hybrid Steps	Nb. of derived variables
Development	41	Subset Ratio	33	Development & MMF	21
Normalization	50	Ratio	17	Development & Subset Ratio	46
Binning	4	Interaction	7	Dichotomization & Binning	8
Dichotomization	1	Mathematical Functions (MMF)	15	Ratio & UMF	7
Mathematical Functions (UMF)	1				
Total	97	Total	72	Total	82

step *ratio* resulted in 17 variables. We used financial, ratio-scaled variables to derive new variables, for example ratio of transaction volumes to assets, ratio of transaction volumes to age, and ratio of electronic and written transfers. Seven variables were derived as *interactions*. Interactions included among others, combinations of region and income classes or age classes and profession. Within the step of *multivariate mathematical functions* we derived 15 new variables, e.g. the amount of products within a division. Another application was the calculation of the ratio between customers' assets and the median of assets over all customers. In this way, we could identify customers with special behavior.

Finally, we derived 82 *hybrid* variables as combination of steps. Most of the hybrid variables are developments of subset ratios (46 variables) or multivariate mathematical functions (21 variables) to illustrate structural changes in the customers' assets. These variations show especially well the changes in the customers' life and needs. Eight metric variables were firstly put into groups and got flagged afterwards. This combination of binning and dichotomization can be promising if only a subset of the original values is relevant for the target classification. The ratio of two mathematical functions was applied for 7 variables. In that context, the logarithmic function was used to make two different dimensions comparable. The logarithm of income divided by the logarithm of age for instance is a good predictor when it comes to target group characterization as it combines two important information about the customers. As the combination of steps is the most complex way to reveal underlying customer behavior, we expect these hybrid variables to be the most powerful in improving classifier performance.

3.2 Results

Finally, 11 derived predictor variables were integrated within the decision tree. These variables can assigned to the following approaches: 7 univariate or hybrid developments, 2 MMF, 1 interaction and 1 ratio & MMF. Figure 4 displays both, the decision tree of the REF and the ALL classifier. The circles in b) show the nodes, where a derived variable is responsible for the split. Four derived variables are making the second split, which indicates a great importance for the overall model. These variables are: changes in the amount of products over 1 and 3 months, changes of bonds in the deposit in the last month and trend of the amount of debit transactions within the last 6 months.

The comparison of gain performance of the two classifiers is displayed by a gains chart (see Figure 5). The reference model (REF) is represented by the lower curve; the test model (ALL) refers to the upper curve. The diagonal line plots the expected response for the entire sample if the classifier is not used. In reference to Figure 5 it can be stated, that the use of derivation methods improves classifier performance as the test model clearly outperforms the reference model for the first decile. The steeper the curve, the higher the gain. The curved line indicates how much one can improve the response by including only those customer who rank in the higher percentiles based on gain. In this study, including the top 10% customers might net more than 95% of the positive responses for the classifier

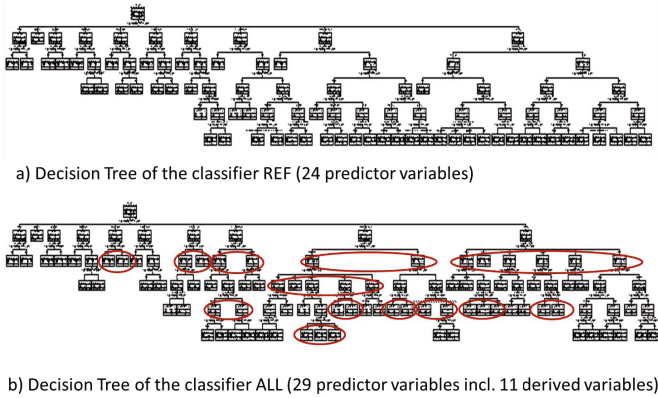


Fig. 4. Decision Trees for Test Set [Source: PASW Modeler 14]

ALL in contrast to about 90% for the classifier REF. This result reveals, that a direct mailing campaign based on the customer classification of the test model (ALL) reaches more "reactors", which means more potential "buyers" of closed funds than the classification of the standard model (REF). The results gained from the decision tree support the following statements:

- The effectiveness of each derivation step depends on sector and target selection. For our target variable closed funds, especially *univariate or hybrid developments* lead to very powerful predictors for classification. Most likely this is due to the fact that they express best the customers' asset structure and financial situation as the following insight outlines. The derived variable

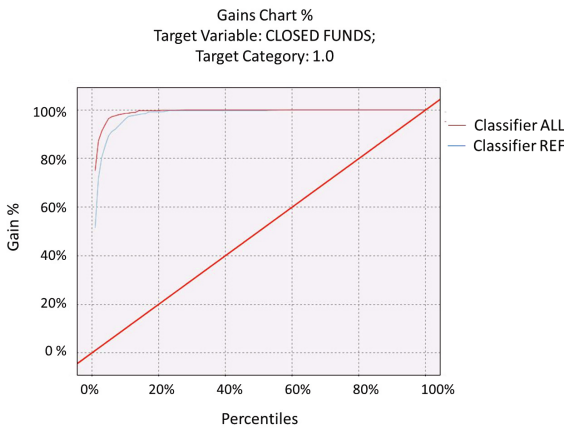


Fig. 5. Gains Chart for Test Set [Source: PASW Modeler 14]

”trend of the amount of debit transactions within the last 6 months” represents the most important variable for the classifier as it discriminates well between reactors and non-reactors.

- The amount of products bought by the customer and the change of this amount over 1 and 3 months are important predictors for the target variable. High values of these variables indicate a high buying activity, possibly due to customer’s satisfaction.
- Young customers with relatively high income holding bonds are potential buyers of closed funds.
- In this study we applied 9 different derivation steps and 4 combinations of steps (see Table 2), from which finally 11 derived variables were integrated into the test model. This shows that applying various derivation approaches is a good way to find the best classifier. Nevertheless, one should bear in mind that too many variables may affect other algorithms more than it happened to the decision trees in our study.

4 Conclusion, Limitations and Future Directions

By setting up a detailed framework we intended to facilitate knowledge-driven data preparation in classification (and other data mining) projects. This hierarchical structured toolkit gives a good methodological overview and turned out to be a powerful guideline when mapping it with business and data know-how. A wide knowledge of statistical approaches becomes less important. However, the principal aim of this study was to empirically show how classifier performance can be enhanced by knowledge-driven data preparation. Therefore, approaches for variable derivation as a subtask of data preparation were conducted and tested by a specific research design. By comparing the gains curve of two classifiers (with and without derived variables), it can be stated that derivation of new variables clearly improves classifier performance. With regard to variable derivation knowledge-driven acting reduces the risk of creating a vast amount of variables, which potentially affect algorithm’s efficiency and accuracy without providing added value of classifier performance. Automated derivation would increase this risk.

This study has also certain limitations. The presented framework can only perform as a guideline and needs to be specified by the user for individual application. With regard to the implementation of derivation methods, it has to be considered that only one dataset from a specific sector has been tested. Additional or other effects on another data structure are possible. Thus, these findings cannot be extrapolated to all datasets. Further limitations exist in terms of applied software and classification algorithm. Our results refer to an application with decision trees employed by PASW Modeler 14. The applied data preparation methods have possibly great potential using neural networks as well. However, more research on this topic needs to be undertaken to stabilize and specify the hypothesis that knowledge-driven data preparation is worthwhile.

Nevertheless, deriving new variables should always lead to great attention in variable selection. As the algorithm’s speed and accuracy can suffer from a

large amount of input variables, further research in the area of variable selection and its combination with variable derivation is necessary. Further research is needed to identify influential factors on the procedure as a first step and to test their influence on the classifier performance as a second step. Possible research areas could be the role of the target variable or the influence of the test design, i.e. do we get a different outcome by applying separate, all or only a specific combination of derivation approaches?

References

1. Rexer, K.: 5th Annual Data Miner Survey - 2011 Survey Summary Report. Rexer Analytics, Winchester (2011)
2. KDnuggets, Which methods/algorithms did you use for data analysis in 2011?, <http://www.kdnuggets.com/polls/2011/algorithms-analytics-data-mining.html>
3. Fayyad, U., Piatesky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.): Advances in Knowledge Discovery and Data Mining. AAAI Press, California (1996)
4. SAS: From Data to Business Advantage: Data Mining, SEMMA Methodology and the SAS System. White Paper, SAS Institute Inc. (1997)
5. Reinartz, T.: Focusing Solutions for Data Mining: Analytical Studies and Experimental Results in Real-World Domains. Springer, Heidelberg (1999)
6. Chapman, P., Clinton, J., Kerber, R., Khabaza, R.T., Reinartz, T., Shearer, C., Wirth, R.: CRISP-DM 1.0: step-by-step data mining guide. SPSS Inc. (2000)
7. Kurgan, L.A., Musilek, P.: A survey of knowledge discovery and data mining process models. *The Knowledge Engineering Review* 21(1), 1–24 (2006)
8. Refaat, M.: Data Preparation for Data Mining Using SAS. Morgan Kaufmann, San Francisco (2007)
9. Anand, S.S., Bell, D.A., Hughes, J.G.: The role of domain knowledge in data mining. In: 4th Int'l ACM Conference on Information and Knowledge Management, pp. 37–43. ACM, New York (1995)
10. de Oliveira Lima, E.: Domain Knowledge Integration in data mining for churn and customer lifetime value modelling: new approaches and applications. Dissertation, University of Southampton (2009)
11. Kopanas, I., Avouris, N.M., Daskalaki, S.: The Role of Domain Knowledge in a Large Scale Data Mining Project. In: Vlahavas, I.P., Spyropoulos, C.D. (eds.) SETN 2002. LNCS (LNAI), vol. 2308, pp. 288–299. Springer, Heidelberg (2002)
12. Sinha, A.P., Zhao, H.: Incorporating domain knowledge into data mining classifiers: An application in indirect lending. *Decision Support Systems* 46, 287–299 (2008)
13. Pyle, D.: Business Modeling and Data Mining. Morgan Kaufmann Publishers, Amsterdam (2003)
14. Linoff, G.S., Berry, M.J.A.: Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management. Wiley Publishing, Indianapolis (2011)
15. Han, J., Kamber, M., Pei, J.: Data Mining, Concepts and Techniques. Morgan Kaufmann, Waltham (2012)
16. Azevedo, A., Santos, M.F.: KDD, SEMMA and CRISP-DM: A Parallel Overview. In: Proceedings of the IADIS European Conference Data Mining, pp. 182–185 (2008)
17. Nisbet, R., Elder, J.F., Miner, G.: Handbook of Statistical Analysis and Data Mining Applications. Academic Press, Elsevier, Amsterdam, Boston (2009)

18. CRISP-DM 2.0 Special Interest Group (SIG), <http://www.crisp-dm.org/new.htm>
19. Hernández, M.A., Stolfo, S.J.: Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery* 2(1), 9–37 (1998)
20. Wang, R.Y., Strong, D.M.: Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of Management Information Systems* 12(4), 5–33 (1996)
21. Rahm, E., Do, H.H.: Data Cleaning: Problems and Current Approaches. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 23(4), 3–26 (2000)
22. Michalski, R.S.: Pattern Recognition as Knowledge-Guided Computer Induction. Technical Report No. 927. Department of Computer Science, University of Illinois, Urbana-Champaign, IL (1978)
23. Wnek, J., Michalski, R.S.: Hypothesis-driven constructive induction in AQ17: A method and experiments. In: *Proceedings of the International Joint Conference on Artificial Intelligence, Workshop on Evaluating and Changing Representations in Machine Learning*, pp. 13–22 (1991)
24. Hammer, M., McLeod, D.: The semantic data model: a modelling mechanism for data base applications. In: Lowenthal, E.I., Nell, B.D. (eds.) *Proceedings of the 1978 ACM SIGMOD International Conference on Management of Data*, Austin, Texas, pp. 26–36 (1978)
25. Matheus, C.J., Rendell, L.A.: Constructive Induction on Decision Trees. In: Sridharan, N.S. (ed.) *11th International Joint Conference on Artificial Intelligence*, pp. 645–650. Morgan Kaufmann (1989)
26. Zheng, Z.: Constructing New Attributes for Decision Tree Learning. Dissertation, Basser Department of Computer Science (1996)
27. Welcker, L.: Segmentierungsansätze zur Variablenreduktion im Rahmen der Optimierung von Scoring-Ergebnissen. Master Thesis, unpublished, Münster University of Applied Sciences (2010)
28. Michie, D., Spiegelhalter, D.J., Taylor, C.C.: *Machine Learning, Neural and Statistical Classification*. Englewood Cliffs, Ellis Horwood (1994)
29. Lim, T.-J., Loh, W.-Y., Shih, Y.-S.: A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms. *Machine Learning* 40, 203–229 (2000)
30. Kass, G.V.: An exploratory technique for investigating large quantities of categorical data. *Journal of Applied Statistics* 29(2), 119–127 (1980)
31. Biggs, D., de Ville, B., Suen, E.: A method of choosing multiway partitions for classification and decision trees. *Journal of Applied Statistics* 18(1), 49–62 (1991)

CWFM: Closed Contingency Weighted Frequent Itemsets Mining

Eunyoung Park, Younghee Kim, Ieejoon Kim, Jaeyeol Yoon,
Jiyeon Lim, and Ungmo Kim

Department of Computer Science and Engineering, Sungkyunkwan University,
300 Chunchun-Dong Jangan-ku, Suwon,
Gyeonggi-do 440-746 Republic of Korea
{wjbest527, younghee}@gmail.com,
{uk3080789, vnt1lff1, 01039374479}@naver.com, umkim@ece.skku.ac.kr

Abstract. Weighted pattern mining have been studied the importance of items. So far, in weight constraint based pattern mining, the weight has been considered the item's price. The price considered as the weight has a limit. The weight characteristic of weighted pattern mining should be considered case-by-case situation. Thus, we motivate by considering the special and individual case-by-case situation to find the exact frequent patterns. We propose how to set weight into frequent patterns mining with a case-by-case condition, called CWFM (closed contingency weighted pattern mining). Moreover, we devise information tables by using statistical and empirical data as strategic decision. In addition, we calculate the contingency weight using outer variables and values which are from information tables. CWFM extracts more meaningful and appropriate patterns reflected case-by-case situation. The proposed new mining method finds closed contingency weighted frequent patterns having a significance which represents the case-by-case situation.

Keywords: closed weighted frequent patterns mining, information table, contingency weight, contingency weighted frequent patterns, outer-variables.

1 Introduction

Most algorithms use a support constraint to prune infrequent patterns and to reduce search space [4-7]. It is difficult for user to decide support value efficiently. One of the main limitations of the previous algorithms are treated all the items uniformly, but real items have different importance than other items [8-13]. For this reason, weighted frequent itemset mining algorithms [8-13] have been suggested that some items should be given priority, such as WFIM [8] is used to the price factor of item as priority. However, it does not consider different factors for using weight. Here, it is necessary to explain factor in connection with weight. If special situation is considered, the substantial results can be obtained frequent pattern considering the situation. In addition to the price factor, weights can be determined by other factors in a wide of dataset. That is, there are climate, area, age, wedding season, national holidays, gender and countries etc. For instance, sales

volume of a thermal underwear and bikini are influenced by season. Therefore, other factors except price may be various and can make more exact mining result.

In this paper, we consider to find the weighted frequent patterns with weighted factor. The proposed algorithm permits to specify the weight considering case-by-case situation, called CWFM. The weight value of item has differences among data, so that the process to determine the weight is needed before mining. It has two steps. First, Information tables: these collect information by using the substantial and empirical data from reliable institution. For example, National Statistical Office, Child Research Institute, Criminal & Investigation Laboratory and answer of survey etc. Secondly, Outer-variables: these are received data to reflect the special situation.

The chosen values from information tables compose a matrix. A determinant of matrix calculates by using the gauss elimination method. The calculated result value is called the contingency weight. Our main goal is to push the contingency weight constraints into the pattern growth algorithm. The main contributions of this paper for the real dataset are: 1) introduction of concept to provide the contingency weight based on the practical situation, 2) description of mining technique in closed frequent pattern mining with contingency weight, 3) implementation of our algorithm, CWFM.

The remainder of this paper is organized as follows: In section 2, we describe problem definition and related work. In section 3, we developed our proposed method and algorithm for weighted contingency frequent patterns. In the section 4, we present performance comparison. Finally, conclusions are given in Section 5.

2 Related Work and Problem Definition

2.1 Related Work

Weighted frequent patterns mining algorithms [8-13] based on the pattern growth method[4] improved the problems of the support constraints with weight constrains. The first, WFIM[8] suggested the method in which normalized weights with a weight range. The closed frequent pattern mining indicates the same knowledge of patterns. If a pattern X is closed frequent pattern mining, there are not proper superset X' in the transaction. The weight value of item has differences among data, so that the normalization process is needed. First, weights of items are given weight range table. According to items' importance, weight of items are given in $w_{(i)\min} \leq w_{(i)} \leq w_{(i)\max}$ for the item. The weight of a pattern is the sum of items divided by length of pattern. Maximum weight (MaxW) is defined the value of maximum weight among items in transaction. The weighted support of a pattern (WS(X)) is the value of multiplying the weight of pattern with the support of pattern. The weighted support is defined such as: $WS(X) = \text{Weight}(X) * \text{Support}(X)$ and it should be grater than \min_sup . Previous studies did not consider unique circumstance as the factor of weight. This paper can describe mining method considering contingency weight in special situation.

2.2 Problem Definition

Let $I = \{i, i, \dots, i_n\}$ be a distinct set of items. A transaction database TDB is a set of transactions in each transaction. A A transaction is denoted as a tuple $\langle tid, X \rangle$. The

tid indicates a unique transaction identifier. The $X = \{x_1, x_2, \dots, x_m\}$, $x_i \in I$, for $1 \leq i \leq m$, is a set of items, while m is called the length of itemset. An itemset is ordered list, $X \in I(2^n - 1)$ where 2^n is the power set of I . An itemset $= \{x_1, x_2, \dots, x_n\}$ is also represented as $x_1 x_2 \dots x_n$. An item is called a k -itemset if it contains k items. The support of a itemset X is the number of transactions containing X in the database. A weight of an item is importance or priority of item. The weighted frequent pattern mining is to find the complete set of patterns satisfying a support threshold and a weight constraint in the database. The closed frequent pattern is superset X' if the support of pattern X equals to that of X' and the length of X is less than that of X' and every transaction containing X also contains X' . An example of the support constraint [1],[5],[12] is the support constraint. On the other hand, a pattern X is infrequent pattern, then, super patterns of the pattern must be infrequent patterns. Hence, infrequent patterns can be pruned to reduce search space.

Table 1. Transaction of database TDB

TID	Set of items	Frequent item list(min_sup =3)
100	a, e, h, g, u	a, e, h, g
200	a, h, g, n, u	h, g, a, n
300	e, g, n, t	g, n, e
400	a, e, h, g, t, w	h, g, a, e
500	h, g, e, w	h, g, e
600	e, g, n	g, n, e

Example 1. Given a transaction database TDB in the Table 1. We have six transactions and 8 items: $\langle a, e, h, g, n, t, u, w \rangle$. Suppose that we have minimum support = 3. A frequent list is: $\langle a:3, e:5, h:4, g:6, n:3 \rangle$. Items “t”, “u” and “w” are pruned because the support of these items is less than a min_sup. A super pattern “hga” is a closed frequent pattern because the support (3) of the pattern is equal to the support (3) of a pattern “ga” and the length(2) of “ga” is less than that(3) of “hga” and every transaction containing “ga” also contains “hga”.

3 CWFM

In this Section, we suggest the CWFM algorithm with the concept of contingency weight. The main approach of CWFM is to push contingency weight into the closed weighted frequent patterns mining algorithm based on the pattern growth method and prune uninteresting patterns. In CWFM, a new measure of weight is defined and related properties are described. Sequential patterns mining have been treated uniformly, but real items or datasets have different importance in special situation. We present our algorithm in detail and give statistical examples to explain the adaptation of contingency weight in the FP-tree construction, then show the projected FP-tree using by bottom up traversal of FP-tree.

3.1 Preprocessing

The preprocessing is to collect the necessary data in the actual situation. Before the mining, the preprocessing is needed. We will explain the health consultation in order to discover the association between diagnosed illness and other diseases. There are information table and outer-variable. The values of information tables can be found by using in the real world statistical data and will be used as the weight value.

Proposal 1. Information tables: These can be made usefully depending on the situation in each dataset and the number of information tables can vary. We collect the data that is associated with the characteristics of dataset. Here, the information tables are based on 2009 year data by Korea Statistical Information Services and are statistical data about deaths and illnesses. Our study consists of three information tables to explain the health consultation. One of them is total death rate table, other is death rate table for the generations and the other is death rate for smoking table. One of created tables should be set a standard table because value of item in standard table is used to header's weight of FP-tree. We will consider the death rate table(table 2) as the standard table. All of information tables, table1, table2, table3, table4, are shown below.

Table 2. Total death rate table

item	a	E	h	g	n	t	u	w
rate of death	0.41	1.61	0.06	0.19	0.9	1.05	0.39	0.26

Table 3. Death rate for the generations table

age item	a	e	h	g	n	t	u	w
0 ~ 9	9.76	8.51	14.92	8.48	30.55	34.83	11.23	11.42
10~19	6.52	5.68	9.97	5.68	20.41	23.27	7.5	7.64
20~29	6.54	5.70	10.0	5.69	20.46	23.34	7.53	7.65
30~39	6.56	5.72	10.06	5.73	20.54	23.46	7.57	7.70
40~49	6.53	5.69	10.14	5.78	20.65	23.63	7.62	7.77
50~59	6.46	5.41	10.20	5.91	20.82	23.92	7.67	7.94
60~69	6.19	4.63	9.93	6.15	21.12	24.44	7.67	8.28
70~79	5.52	3.44	8.48	6.69	21.73	24.81	7.31	9.04

Each item in the tables indicates name of a disease. Then, item “a” is stomach cancer, “e” is liver cancer, “h” is lung cancer, “g” is hypertensive heart disease, “n” is a cardiac disorder, “t” is a cerebrovascular disorder, “u” is diabetes mellitus and “w” is pneumonia in the tables. In 2009, the death toll is 246,942 persons (then, crude death rate of 497.3 per 100,000). The term “element” can be defined as value in the information table. The element of each item refers to the mortality of each disease in tables. For example, the element of item “e” represents 1.61 in the table 2. Every

1-row in tables indicates names of disease. The 1-columun in the table 3 means the generations. Finally, 1-columun in table 4 means the number of cigarette smoking per a day.

Table 4. Death rate for smoking table

num \ item	a	e	h	g	n	t	u	w
0~9	34	6.2	7.4	0.55	1.89	5.44	0.77	0.75
10~19	1.98	5.4	6.5	0.45	1.26	3.56	0.23	0.63
20~29	1.85	4.32	1.12	0.23	1.12	3.45	0.27	0.34
30~39	1.87	4.33	1.23	0.25	1.13	3.66	0.29	0.33
40~49	1.98	5.4	1.50	0.32	1.26	3.67	0.33	0.36
50~59	31.27	5.71	1.73	0.45	2.41	5.34	0.66	0.5
60~69	4.02	3.86	2.16	0.68	2.26	7.16	0.85	0.64
70~79	2.09	1.19	2.09	0.48	1.27	5.63	0.48	0.64

Proposal 2. Outer-variables: It may be determined by reflecting the situation. In other word, there are items that represent the situation among items. The item values matching outer variables are selected in each table as an element of the matrix. For instance, assume that there is one man who is 43 years old and suffered from pneumonia in the past, and smokes 10 cigarettes per a day. Here, the outer variables set up 43 years old, pneumonia, 10 cigarettes. Here, outer variables are ‘e’ , 43 and 10.

3.2 Contingency Weight

As shown in the above method, the values extracted from each table become elements of square matrix. The determinant of matrix computes by using the Gauss elimination method and is defined $\det(\text{Matrix})$. Then, we are set the value of ‘1’ at empty elements. Because the value of 1 does not influence any other number even if multiply any number by 1. The $\det(\text{Matrix})$ is multiplied to value of items in the standard table matching outer variables, respectively. At this time, calculated result value is called the contingency weight. Therefore, the weight can be increased by considering case-by-case situation as weight factor. The item value from other tables matching outer variable are extracted. Then, these constitute the matrix. The determinant of matrix computes using the Gauss elimination method. That is, it is way to increase the importance of item related special situation.

Definition 1. [Contingency Weight (CWeight)] There are given weight of item matching outer-value(E) in the standard table and the value of determinant of matrix $\det(\text{matrix})$. Three tables made the matrix with row 3 and column 3.

The contingency weight is defined as follows.

$$\text{CWeight (E)} = \text{weight (E)} * \det(\text{matrix}).$$

Example 2. Assume that outer variables are entered item “a” and item “e”, additionally, age “43” and the quantity of smoking per a day “10” are entered. The item “a” selects “0.41” and the item “e” picks “1.61” in the standard table (Table 2), respectively. Outer variables are treated ascending order in items. The each value becomes element of the matrix as 1-row 1-column and 1-row 2-column. Next, the 1-row 3-column is set “1” as the default value. Similarly, by using “43” entered as an outer variable, 43 and the item “a” intersect at the point value “6.53”, “43” and the item “e” intersect at the point value “5.69” in the death rate for the generations table. Finally, the item “a” and “10” intersect at the point value “1.98”, the item “e” and “10” intersect at the point value “5.4” in the smoking table . As a result, the 3 x 3 matrix is constructed.

$$|Matrix| = \begin{vmatrix} 0.41 & 1.61 & 1 \\ 6.53 & 5.69 & 1 \\ 1.98 & 5.4 & 1 \end{vmatrix}$$

Fig. 1. Matrix

According to the Gauss elimination method, the matrix changes as fig 2.

$$|Matrix| = \begin{vmatrix} 0.41 & 1.61 & 1 \\ 0 & -19.95 & 14.93 \\ 0 & 0 & -6.61 \end{vmatrix}$$

Fig. 2. Result Matrix

Fig 2 is upper triangle matrix. The determinant of det (matrix) is about 54.07. So, item “a” and item “e” in the standard table(table2) are multiplied by 54.07 respectively. That is, a’s and e’s contingency weight are 22.17 and 87.05 . The weight of item “a” is increased from “0.41” to “22.17” and the weight of item “e” is changed from “1.61” to “87.05” in the standard table.

3.3 Contingency Weight Pattern

The frequent weight pattern found by using the Definition 1 means the contingency pattern. Our algorithm reflects to the weights by considering the characteristic of dataset and after collecting the empirical data. The different way of increase the importance of item proposes. There is strong point that user can get proper and accurate information.

3.4 FP (Frequent Pattern) Tree Structure

CWFP uses FP-trees as a compress structure and are used in pattern growth algorithms. CWFP computes local frequent items of a prefix by scanning its projected database once. The Contingency weight(CWeight) increases the weight of item matching outer-variables. So, CWeight is greater than weight in the standard table.

After this, sort the items in weight ascending order. A header table of FP-tree has four fields: "item-id", "weight", "support" and "node-link". The first field, Item-id is distinct item. The second field, weight indicates weight of items including contingency weight(CWeight). The third field, support is used to count item in the transaction. The last field, node-link is linked to child-node with same item in the FP-tree. Top of FP-tree has only a root node. The first node of root in the FP-tree has item-id and pointer. When an item is inserted in the FP-tree, sort the items in contingency weight ascending order. For instance, diseases are "a", "e", "n" and age is "43" and the quantity of smoking is 10 as outer variables. The item "a" selects "0.41" and the item "e" picks "1.61" and the item "n" select "0.9" in the standard table(table 2). The each value becomes element of the matrix as 1-row 1-column, 1-row 2-column and 1-row 3-column. In table 3, by using "43", 43 and the item "a", "e" and "n" intersect at the point value "6.53", "5.69" and "20.65", respectively. Finally, each item "a", "e", "n" and "10" intersect at the point value "1.98", "5.4" and "1.26" in table 4. As show in Fig 2, a result table, the matrix is constructed.

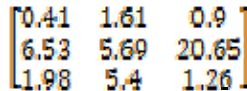


Fig. 3. Matrix

The determinant of matrix is 25.29. As discussed above, a f_list is: <a:3, e:5, h:4, g:6, n:3 > from table 1. Items "p", "u" and "w" are pruned since the support of them are less than a min_sup (3). Before construct a header of FP-tree, contingency weight is computed. That is, weight list are <a:10.37, e:40.72, h:0.06, g:0.19, n:22.76 >. So, the weighted frequent items are sorted by the weight ascending order, {(h, g, a, e), (h, g, a, n), (g, n, e), (h, g, a, e), (h, g, e), (g, n, e)} in each transaction. The sorted weighted frequent items in the each transaction are sequentially inserted in a FP-tree along a path from the root to the corresponding node. If a new node is inserted in the path, only the support of the node is set to 1. If an existing node in the FP-tree is used, the support of the node is increased by one. Fig 4 presents the FP-tree and a corresponding header table.

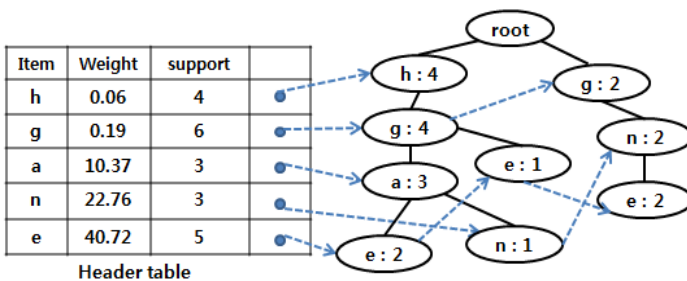


Fig. 4. Global FP-tree for CWFM

FP-tree with the contingency weight mine closed frequent itemsets by adapting bottom up traversal. FP-tree mines the patterns including item “e” which has the highest weight. First, e’s conditional database is generated by starting from e’s head and following e’s node-link. The conditional database for prefix “e” contains three transactions: {“hga:2”, “hg:1”, “gn:2”}. Item “a” and item “n” are pruned because item a’s support (2) and item n’s support (2) are less than a minimum support (3). A local conditional FP-tree with the prefix “e” are “hge:2”, “hg:1” and “gn:2”. All kinds of combinations of item related to the prefix “e” are “e:5”, “he:3”, “ge:5” and “hge:3”. The patterns “e:5” and “he:3” are pruned since the contingency weighted frequent pattern “ge:5” and “hge:3” are superset with same support according to closer property, respectively. As a result, closed contingency weighted frequent patterns for the prefix “e” are <ge:5>, and <hge:3>. It is shown fig 5 (a).

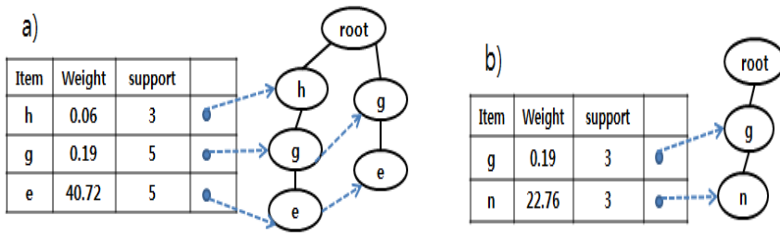


Fig. 5. Local conditional with the prefix “e” (a) and “n” (b)

From second the local conditional FP-tree with the prefix “n”, we can find out the contingency weighted frequent patterns are “hga:1” and “g:2”. We get closed contingency weighted frequent patterns related to “n:3” : <gn:3 >. So, it is closed contingency weighted pattern. Then, we can construct local conditional FP-tree from the FP-tree and mine closed contingency weighted frequent pattern from them recursively. Closed contingency weighted frequent pattern with the prefix “a” is <hga:3> and with the prefix “g” is <hg:4> and with the prefix “h” is empty.

3.5 CWFM Algorithm.

The algorithm for the CWFM is described as follows: Initially, the CWFM scans TDB once to find the f_list. Then, considering characteristic of dataset, analyze what data should be used to it and set outer-variables. This consist information tables and outer-variables. Secondly, a matrix is constructed weight of item in each table matching outer-value and det(matrix) is computed using the Gauss elimination method. Then, the contingency weight obtains, to adopt weight of item matching outer-value in standard table, value of multiplying item’s value with det(matrix). Thirdly, a global FP-tree is constructed from weight of f_list, in weight ascending order. Finally, contingency weighted frequent mining is performed.

CWFM Algorithm**Input :**

- D** : A transaction database (TDB),
- δ** : minimum support threshold : min_sup,
- \tilde{o}** : outer variables,
- \mathcal{I}** : information tables from the national statistical office

Output :

CWFP: The complete set of closed contingency weighted frequent pattern.

Method :

- 1) Generator in $D1 \leftarrow \{1\text{-itemsets}\}$; // $D1 \leftarrow$ support-itemset (D1);
- 2) for each generator $p \in D1$ do begin
- 3) if (support(p) < δ) then delete p from D1;
- 4) else $F_list \leftarrow p$; // F_list : frequent item list of D1
- 5) end
- 6) $\tilde{o}_{[m]} \leftarrow$ Input outer variables
- 7) $max = m$; // $\tilde{o}_{[m]} \rightarrow$ here, m is number of outer variable
- 8) for ($i = 1$; $i \leq max$; $i++$) $\mathcal{I}[i]$; //information tables create
- 9) end
- 10) for($y=1$; $y < number\ of\ \tilde{o}$; $y++$) // matrix create
- 11) for($z = 1$; $z < number\ of\ \tilde{o}$; $z++$)
- 12) for ($i = 1$; $i \leq max$; $i++$) //for each information table
- 13) for($n = 0$; $\tilde{o}_{[n]} \leq max$; $n++$)
- 14) for($row = 1$; $row \leq \mathcal{I}[i].length$; $row++$)
- 15) if ($\mathcal{I}_i[row] = \tilde{o}_{[n]}$) $v = row$;
- 16) end if
- 17) end
- 18) end
- 19) for($n = 0$; $\tilde{o}_{[n]} \leq max$; $n++$)
- 20) for($col = 1$; $col \leq \mathcal{I}_i[i].length$; $col++$)
- 21) if ($\mathcal{I}_i[col] = \tilde{o}_{[n]}$) $w = col$;
- 22) end if
- 23) end
- 24) end
- 25) $matrix[y][z] = \mathcal{I}_k[v][w]$;
- 26) end
- 27) end
- 28) end
- 29) $det(matrix) \leftarrow matrix$;
- 30) for each $o \in \tilde{o}_{[i]}$ do begin //calculate contingency weight
- 31) if ($p = o$) then $\hat{c} = p$ (in standard table) * $det(matrix)$;
- 32) end if

Fig. 6. Pseudo code for the CWFM

- 33) end
- 34) Weight (p) $\leftarrow \hat{c}$ // \hat{c} is contingency weight
- 35) Header's weight \leftarrow Sort items in contingency weight ascending order.
- 36) Call call FP-tree(f-list);
- 37) Call Conditional FP-tree (FP-tree, { }, CWFM);

Fig. 6. (continued)

4 Experimental Results.

4.1 Performance Comparison

In this section, we present the performance of our CWFM algorithm by using contingency weights. We also compared CWFM with WFIM[8]. CWFM was written in Java. Experiments were performed on a 366Mhz Pentium PC with 512MB main memory, running on Microsoft Windows/XP. WFIM adjusts by setting weight ranges to reduce the number of frequent itemsets when giving weights to each item.

The main purpose of experiments is to certain how efficiently the weighted frequent pattern can be found by using contingency weight. In this performance test, number of information tables and number of pattern are checked. Fig 6 and Fig 7 show the results of performance evaluation based on the connect dataset which contain 100k to 700k transactions. This dataset is real dataset and have 43items in the each transaction.

Fig 6 illustrates that performance when using and CWFM is better than using WFIM.

Although the number of information tables increase, Fig 7 also show that the slope of CWFM is lower than that of WFIM. CWFM sees somewhat better scalability than WFIM.

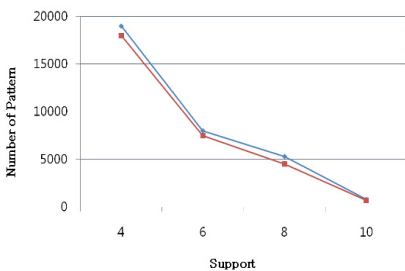


Fig. 7. number of patterns

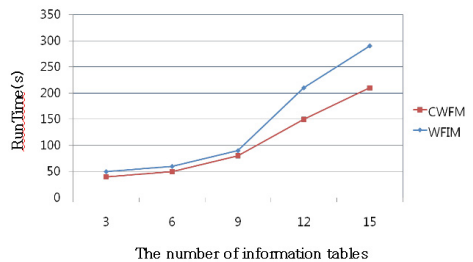


Fig. 8. Runtime for the CWFM

Table 5. The number of frequent patterns by increasing outer_variable

min_support	Num of the outer_variables	Num of contingency weight pattern
54046(80%)	10	3,565
	20	935
	30	203

Table 5 lists the difference of number of frequent patterns by increasing the number of the outer-variables. It shows that the number of frequent patterns decrease considerably by increasing the number of outer_variables or information table. Of course, the values in the information table and the value of the contingency weight (CWeight) that reflects the real world. Therefore, the number of patterns may be different.

5 Conclusion

In this paper, we studied the problem of weight factor in weighted frequent pattern mining. In previous studies, the items are used prices as weight factor. However, the factors of items are various in dataset or application. We introduced information tables and the concept of contingency weighted pattern by using statistical and empirical data. CWFm focuses on closed weighted frequent pattern by considering special situation. To assign the weights of items in special dataset, outer-variables are emphasis. Hence, the contingency patterns are considerably valuable for considering special situations. In the forward, weighted pattern mining should be considered opinion of experts and the number of information table. This is to reflect substantial part in terms of information. The weights based information tables can be support to extract more accurate prediction information in the case-by-case situation and valuable patterns by adopting the contingency weight.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD 1993 (1993)
2. Agrawal, R., Srikant, R.: Mining Sequential Patterns. In: ICDE 1995 (1995)
3. Srikant, R., Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) EDBT 1996. LNCS, vol. 1057, pp. 1–17. Springer, Heidelberg (1996)
4. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM SIGMOD 2000 (May 2000)
5. Pei, J., Han, J.: CLOSET: An Efficiently Algorithm for Mining Frequent Closed Itemsets. In: DMKD 2000 (May 2000)
6. Pei, J., Han, J., Mortazavi-Asi, B., Pino, H.: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix Projected Pattern Growth. In: ICDE 2001 (2001)

7. Wang, J., Han, J., Pei, J.: CLOSET+: searching for the best strategies for mining frequent closed itemsets. In: ACM SIGKDD 2003 (August 2003)
8. Yun, U., Leggett, J.J.: WFIM: Weighted Frequent Itemset Mining with a weight range and a weight range and minimum weight. In: SDM 2005 (April 2005)
9. Yun, U., Leggett, J.J.: WLPMiner: Weighted Frequent Pattern Mining with Length-Decreasing Support Constraints. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 555–567. Springer, Heidelberg (2005)
10. Yun, U., Leggett, J.J.: WIP: Weighted Interesting Patterns with a strong weight and/or support affinity. In: SDM 2006 (April 2006)
11. Yun, U., Leggett, J.J.: WSpan: Weighted Sequential Pattern mining in large sequence databases. In: Proc. of the Third Int'l Conf. on IEEE Intelligent Systems (September 2006)
12. Yun, U.: Mining Lossless Closed Frequent Patterns with Weight Constraints. Knowledge Based Systems 20, 86–97 (2007)
13. Yun, U.: WIS: Weighted Interesting Sequential Pattern Mining with a Similar Level of Support and/or Weight. ENRI Journal 29(3) (June 2007)

Prognostic Modeling with High Dimensional and Censored Data

Leon Bobrowski^{1,2} and Tomasz Łukaszuk¹

¹Faculty of Computer Science, Bialystok University of Technology, Bialystok, Poland
{l.bobrowski,t.lukaszuk}@pb.edu.pl

²Institute of Biocybernetics and Biomedical Engineering, PAS, Warsaw, Poland
leon@ibib.waw.pl

Abstract. Designing linear prognostic models on the base of multivariate learning set with censored dependent variable is considered in the paper. The task of linear regression model designing has been reformulated here as a problem of testing the linear separability of two sets. The convex and piecewise linear (*CPL*) criterion functions are used here both for estimation of the model parameters and for the feature selection task. The feature selection is aimed on neglecting a possibly large amount of independent variables while improving resulting model quality. Particular attention is paid to modeling censored data used in survival analysis. Experiments with the use of the *RLS* method of gene subset selection in prognostic model selection with the censored dependent variable is also described in the paper.

Keywords: linear prognostic models, censored data, high dimensional data, feature selection, *CPL* criterion function.

1 Introduction

Prognostic models are often created with the use the regression analysis methods [], []. Regression analysis includes many techniques focused on modeling the linear relationship between dependent variable and one or more independent variables. In this case, the value of dependent (*target*) variable is predicted as the linear combination of some independent variables. The linear regression function is based on a finite number of unknown parameters that are estimated from the learning data set. The least squares method of the parameters estimation was commonly used in the earliest form of regression analysis [].

The unknown parameters are estimated through minimization of the convex and piecewise linear (*CPL*) criterion functions in the presented work []. The basis exchange algorithms, which are similar to the linear programming, allow to find efficiently the minimal value of the *CPL* criterion functions []. The parameters that create the minimum of the *CPL* criterion function are used in the definition of the optimal prognostic models.

The procedure of prognostic model selection usually involves the feature selection task. Feature selection procedures are aimed at neglecting as large as possible number

of such features which are irrelevant or redundant for a given problem. The resulting feature subset should allow to build a model on the base of available learning data sets that generalizes better to new (unseen) data. The aim is to achieve higher prediction accuracy.

Feature selection problem is particularly challenging in the case when the number of objects in a given database is low in comparison to the number of features used to represent these objects. Such situation appears typically in the case of genetic data sets where the number of features can be thousands of times greater than the number of objects.

Prognostic models developed in the framework of the survival analysis are important in many biomedical applications. In particularly an evaluation new drugs or therapeutic treatments is carried out in accordance with the survival analysis rules. The survival analysis models are designed on the basis of the so called *censored* data sets. The Cox model plays a fundamental role in the survival analysis [1].

The problems related to prognostic model selection (designing) on the basis of interval-censored genetic data is discussed. Designing prognostic models in this case requires, inter alia, a significant reduction of dimensionality. Censored data set could be treated as a special case of interval data set [2]. The possibility of the regression model designing on the basis of genetic data sets with the censored dependent variable by using the convex and piecewise linear (CPL) criterion function is considered. Both theoretical as well as experimental results are described in the paper.

2 Regression Learning Sets with Different Structure

We are considering multivariate regression models based on linear (affine) transformations of n -dimensional feature vectors $\mathbf{x}[n]$ belonging to the feature space $F[n]$ ($\mathbf{x}[n] \in F[n]$) on the points y of the line ($y \in R^1$):

$$y(\mathbf{x}[n]) = \mathbf{w}[n]^T \mathbf{x}[n] + w_0 \tag{1}$$

where $\mathbf{w}[n] = [w_1, \dots, w_n]^T \in R^n$ is the parameters (*weight*) vector and w_0 is the *intercept coefficient* ($w_0 \in R^1$).

Properties of the model (1) depend on the choice of the parameters $\mathbf{w}[n]$ and w_0 . The weights w_i and the threshold w_0 are estimated on regression learning sets. In the case of classical regression analysis the learning sets have the below structure [3]:

$$C_1 = \{\mathbf{x}_j[n]; y_j\} = \{x_{j1}, \dots, x_{jn}; y_j\}, \text{ where } j = 1, \dots, m \tag{2}$$

Each of m objects O_j is characterized in the set C_1 by values x_{ji} of n independent variables (*features*) X_i , and by the observed value y_j ($y_j \in R^1$) of the *dependent (target)* variable Y .

Components x_{ji} of the j -th feature vector $\mathbf{x}_j[n]$ could be treated as the numerical results of n standardized examinations of the given object O_j ($x_{ji} \in \{0,1\}$ or $x_{ji} \in R^1$). Each vector $\mathbf{x}_j[n]$ can be treated also as a point in the n -dimensional feature space $F[n]$.

In the case of *classical regression*, the parameters $\mathbf{w}[n]$ and θ are estimated in accordance with the *last squares method* in such a manner that the sum of the squared differences $(y_j - \hat{y}_j)^2$ between the observed target variable y_j and the modeled variable $\hat{y}_j = \mathbf{w}[n]^T \mathbf{x}_j[n] + w_0$ (1) is minimal []. The optimal regression parameters $\mathbf{w}^*[n]$ and w_0^* (1) can be computed on the basis of the below equation []:

$$\mathbf{b}^*[n+1] = [\mathbf{w}^*[n]^T, w_0^*]^T = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{y}[m] \tag{3}$$

where the *observed target vector* $\mathbf{y}[m] = [y_1, \dots, y_m]^T$ has m components equal to the observed values y_j (2) of the dependent variable Y . The matrix \mathbf{Z} has m rows composed of the *augmented feature vectors* $\mathbf{z}_j[n+1] = [\mathbf{x}_j[n]^T, 1]^T$:

$$\mathbf{Z} = [\mathbf{z}_1[n+1], \dots, \mathbf{z}_m[n+1]]^T \tag{4}$$

In the case of the *interval regression*, an additional knowledge about dependent variable Y of particular objects O_j is represented by the intervals $[y_j^-, y_j^+]$ ($y_j^- < y_j^+$) instead of the exact values y_j (2) [], []:

$$C_2 = \{\mathbf{x}_j[n], [y_j^-, y_j^+]\}, \text{ where } j = 1, \dots, m \tag{5}$$

where y_j^- is the lower bound ($y_j^- \in \mathbb{R}^1$) and y_j^+ is the upper bound ($y_j^+ \in \mathbb{R}^1$) of unknown value y_j ($y_j^- < y_j < y_j^+$) of the target variable Y which, which accompanies the feature vector $\mathbf{x}_j[n]$.

Remark 1: The classical learning set C_1 (2) can be transformed into the interval learning set C_2 (5) by introducing the boundary values $y_j^- = y_j - \epsilon$ and $y_j^+ = y_j + \epsilon$, where ϵ is a small positive parameter (*margin*).

The interval learning set C_2 (5) can represent in a natural manner imprecise measurements of the dependent variable Y . Let us assume for a moment, that the dependent variable Y represents the *survival time* T and has been measured in years. In this case, the j -th measured value of the variable Y is equal to y_j years. If we wish to express the j -th measured value y_j in months we can use the interval $[12y_j, 12(y_j+1)]$.

Prognostic models developed in the framework of the survival analysis are used, inter alia, for prognosis of the survival time T []. Let us assume, that the prognostic model $T(\mathbf{x})$ of an unknown survival time T is linear (1):

$$T(\mathbf{x}) = \mathbf{w}[n]^T \mathbf{x}[n] + w_0 \tag{6}$$

Learning data sets C_s in survival analysis traditionally have the following structure [], []:

$$C_s = \{\mathbf{x}_j[n], t_j, \delta_j\} \quad (j = 1, \dots, m) \tag{7}$$

where t_j is the observed survival time between the entry of the j -th patient O_j into the study and the end of the observation, δ_j is an indicator of failure of this patient ($\delta_j \in \{0, 1\}$): $\delta_j = 1$ - means the end of observation in the event of interest (*failure*), $\delta_j = 0$ - means that the observation of the j -th patient O_j ended before the event. In this case ($\delta_j = 0$) information about survival time t_j is not complete (the *right censored*

observation). The *real survival time* T_j can be defined in the below manner on the basis of the set C_s (7):

$$(\forall j = 1, \dots, m) \text{ if } \delta_j = 1, \text{ then } T_j = t_j, \text{ and} \tag{8}$$

$$\text{if } \delta_j = 0, \text{ then } T_j > t_j$$

Definition 1: The *right censoring* means that an unknown survival time T_j of the j -th patient O_j is **longer** than the observed time t_j . The *left censoring* means that an unknown survival time T_j of the j -th patient O_j is **shorter** than the observed time t_j .

The censored survival times T_j can be represented also by intervals (5) through introducing two numbers (parameters) – the *lower bound* t_j^- ($t_j^- \in R^1$) and the *upper bound* t_j^+ ($t_j^+ \in R^1$), where $t_j^- < t_j^+$. These parameters define the time interval $[t_j^-, t_j^+]$ for an unknown survival time T_j ($T_j \in [t_j^-, t_j^+]$). In the case of the right censoring, an unknown survival time T_j is greater than the given (known) lower bound t_j^- ($T_j > t_j^-$). It can mean, that $T_j \in [t_j^-, +\infty)$. In the case of the left censoring, an unknown survival time T_j is less than the given (known) upper bound t_j^+ ($T_j < t_j^+$). It can mean, that $T_j \in (-\infty, t_j^+]$. In accordance with such data representation, the right censoring means the replacement of the upper bound t_j^+ by the positive infinity $+\infty$. Similarly, the left censoring means the replacement of the lower bound t_j^- by the negative infinity $-\infty$. In the context of the survival time t_j^+ meaning, the more reasonable representation of the left censoring could be $[0, t_j^+]$ ($T_j \in [0, t_j^+]$).

Both the right censored and the left censored times T_j can be represented by using the interval data set C_2 (5) with the below structure:

$$C_4 = \{ \mathbf{x}_j[n], [t_j^-, t_j^+], \delta'_j \} \quad (j = 1, \dots, m) \tag{9}$$

where δ'_j is the *indicator of censoring* of the survival time T_j of the patient O_j ($\delta'_j \in \{-1, 0, 1\}$): $\delta_j = -1$ means the left censoring ($T_j \in [0, t_j^+]$), $\delta_j = 1$ means the right censoring ($T_j \in [t_j^-, +\infty)$), and $\delta_j = 0$ means that $T_j \in [t_j^-, t_j^+]$, where $0 < t_j^- < t_j^+ < \infty$.

The below linear inequalities can be useful for the purpose of the model (6) designing from the survival data set C_4 (9):

$$\text{if } \delta_j = -1, \text{ then } \mathbf{w}[n]^T \mathbf{x}_j[n] + w_0 < t_j^+ \tag{10}$$

$$\text{if } \delta_j = 1, \text{ then } \mathbf{w}[n]^T \mathbf{x}_j[n] + w_0 > t_j^- \tag{11}$$

$$\text{if } \delta_j = 0, \text{ then } t_j^- < \mathbf{w}[n]^T \mathbf{x}_j[n] + w_0 < t_j^+ \tag{12}$$

The term model (6) designing means here finding such parameters $\mathbf{w}[n]$ and w_0 that the above linear inequalities are fulfilled in the best way possible for feature vectors $\mathbf{x}_j[n]$ from the set C_4 (9).

The parameters $\mathbf{w}[n]$ and w_0 of the regression model (6) are typically estimated from the data set C_2 (5) by using the *Expectation Maximization (EM)* algorithms [4], [5]. There are rather troublesome procedures with serious drawbacks concerning among others a low efficiency, particularly in the case of high dimensional feature space $F[n]$.

Here we are examining the problem of prognostic models designing on the basis of the data set C_4 (9) by using the concept of the linear separability [1]. The linear separability of two data sets is evaluated in the paper through the minimisation of the convex and piecewise linear (CPL) criterion functions defined on these sets [].

3 Linear Separability of Survival Learning Sets

Let us take into considerations two data sets: the *positive set* $G^+[n]$ and the *negative set* $G^-[n]$ which are composed of n -dimensional feature vectors $\mathbf{x}_j[n]$ al $(\mathbf{x}_j[n] \in F[n])$:

$$G^+[n] = \{\mathbf{x}_j[n]: j \in J^+\} \text{ and } G^-[n] = \{\mathbf{x}_j[n]: j \in J^-\} \tag{13}$$

where J^+ and J^- are disjointed sets ($J^+ \cap J^- = \emptyset$) of indices j .

Definition 2: The positive set $G^+[n]$ and the negative set $G^-[n]$ (13) are *linearly separable*, if and only if there exists such a weight vector $\mathbf{w}[n]$ ($\mathbf{w}[n] \in R^n$) and a threshold θ ($\theta \in R$), that all the below inequalities are fulfilled:

$$\begin{aligned} (\exists \mathbf{w}[n], \theta) (\forall \mathbf{x}_j[n] \in G^+[n]) \quad \mathbf{w}[n]^T \mathbf{x}_j[n] > \theta \\ \text{and } (\forall \mathbf{x}_j[n] \in G^-[n]) \quad \mathbf{w}[n]^T \mathbf{x}_j[n] < \theta \end{aligned} \tag{14}$$

The parameters $\mathbf{w}[n]$ and θ define the below hyperplane $H(\mathbf{w}[n], \theta)$ in the feature space $F[n]$ ($\mathbf{x}[n] \in F[n]$):

$$H(\mathbf{w}[n], \theta) = \{\mathbf{x}[n]: \mathbf{w}[n]^T \mathbf{x}[n] = \theta\} \tag{15}$$

If all the inequalities (14) are fulfilled, then each feature vector $\mathbf{x}_j[n]$ from the set $G^+[n]$ is situated on the *positive side* ($\mathbf{w}[n]^T \mathbf{x}_j[n] > \theta$) of the hyperplane $H(\mathbf{w}[n], \theta)$ (15) and each feature vector from the set $G^-[n]$ is situated on the *negative side* ($\mathbf{w}[n]^T \mathbf{x}_j[n] < \theta$) of this hyperplane.

The concept of *linear separability* is used from many years in the theory of neural networks and in pattern recognition methods [1]. Among others, the linear separability has been used in the proof of the convergence of the error-correction algorithm – classic learning algorithm of neural networks. The optimal linear classifiers can be designed through exploration of the linear separability of the data sets $G^+[n]$ and $G^-[n]$ (14) [9].

It is convenient to use the *augmented feature vectors* $\mathbf{z}_j[n+1] = [\mathbf{x}_j[n]^T, 1]^T$ and the *augmented vector of parameters* $\mathbf{v}[n+1] = [\mathbf{w}[n]^T, -\theta]^T$ in definition of the linear separability (14) []:

$$\begin{aligned} (\exists \mathbf{v}[n+1]) (\forall \mathbf{x}_j[n] \in G^+[n]) \quad \mathbf{v}[n+1]^T \mathbf{z}_j[n+1] > 0 \\ \text{and } (\forall \mathbf{x}_j[n] \in G^-[n]) \quad \mathbf{v}[n+1]^T \mathbf{z}_j[n+1] < 0 \end{aligned} \tag{16}$$

Definition 3: The sets $G^+[n]$ and $G^-[n]$ (13) are *linearly separable*, if and only if there exists such augmented vector of parameters $\mathbf{v}'[n+1]$, that all the below inequalities are fulfilled:

$$\begin{aligned}
 (\exists \mathbf{v}'[n+1]) (\forall \mathbf{x}_j[n] \in G^+[n]) \quad \mathbf{v}'[n+1]^T \mathbf{z}_j[n+1] \geq 1 \\
 \text{and} \quad (\forall \mathbf{x}_j[n] \in G[n]) \quad \mathbf{v}'[n+1]^T \mathbf{z}_j[n+1] \leq -1
 \end{aligned}
 \tag{17}$$

We can remark, that the *Definition 3* of the linear separability is equivalent to the *Definition 2*. It means that if the linear inequalities (13) are fulfilled then also the linear inequalities (17) are fulfilled and vice versa.

The linear separability (17) of the sets $G^+[n]$ and $G[n]$ (13) typically occurs in the case when the number n of features x_i is larger than the number m of the feature vectors $\mathbf{x}_j[n]$ in these sets [].

The linear inequalities (17) are used in the definition of the convex and piecewise linear (CPL) penalty and criterion functions []. Among others, designing the optimal linear classifiers on the basis of data sets $G^+[n]$ and $G[n]$ (13) can be carried out through the minimization of the CPL criterion functions.

The concept of the linear separability can be enhanced also from the survival inequalities (11), (12) and (13). Let us introduce for this purpose two types of the *augmented feature vectors* $\mathbf{z}_j^+[n+2]$ and $\mathbf{z}_j^-[n+2]$ and the *augmented weight vector* $\mathbf{v}[n+2]$ ($\mathbf{v}[n+2] \in R^{n+2}$):

$$\begin{aligned}
 & (\forall j \in \{1, \dots, m\}) \\
 \text{if } (\delta_j \geq 0), \text{ then } \mathbf{z}_j^+[n+2] &= [\mathbf{x}_j[n]^T, 1, -t_j^-]^T \text{ else } \mathbf{z}_j^+[n+2] = \mathbf{0}, \\
 & \text{and} \\
 \text{if } (\delta_j \leq 0), \text{ then } \mathbf{z}_j^-[n+2] &= [\mathbf{x}_j[n]^T, 1, -t_j^+]^T \text{ else } \mathbf{z}_j^-[n+2] = \mathbf{0}
 \end{aligned}
 \tag{18}$$

and

$$\mathbf{v}[n+2] = [v_1, \dots, v_{n+2}]^T = [\mathbf{w}[n]^T, w_0, \beta]^T, \text{ where } \mathbf{v}[n+2] \in R^{n+2}
 \tag{19}$$

where β is the *interval parameter* ($\beta \in R^1$).

The inequalities (10), (11) and (12) can be represented by using the symbols $\mathbf{z}_j^+[n+2]$, $\mathbf{z}_j^-[n+2]$ and $\mathbf{v}[n+2]$ in the below manner:

$$\begin{aligned}
 (\exists \mathbf{v}[n+2]) (\forall j \in \{1, \dots, m\}) \\
 (\forall \mathbf{z}_i^+[n+2] \neq \mathbf{0}) \quad \mathbf{v}[n+2]^T \mathbf{z}_i^+[n+2] > 0, \text{ and} \\
 (\forall \mathbf{z}_j^-[n+2] \neq \mathbf{0}) \quad \mathbf{v}[n+2]^T \mathbf{z}_j^-[n+2] < 0
 \end{aligned}
 \tag{20}$$

or (17)

$$\begin{aligned}
 (\exists \mathbf{v}'[n+2]) (\forall j \in \{1, \dots, m\}) \\
 (\forall \mathbf{z}_i^+[n+2] \neq \mathbf{0}) \quad \mathbf{v}'[n+2]^T \mathbf{z}_i^+[n+2] \geq 1, \text{ and} \\
 (\forall \mathbf{z}_j^-[n+2] \neq \mathbf{0}) \quad \mathbf{v}'[n+2]^T \mathbf{z}_j^-[n+2] \leq -1
 \end{aligned}
 \tag{21}$$

Let us introduce the *positive set* $Z^+[n+2]$ and the *negative set* $Z^-[n+2]$ which are composed of such $(n + 2)$ - dimensional vectors $\mathbf{z}_j^+[n+2]$ ($j \in J^+$) and $\mathbf{z}_j^-[n+2]$ ($j \in J^-$) which are different from zero (20):

$$Z^+[n+2] = \{\mathbf{z}_j^+[n+2]: j \in J^+\} \text{ and } Z^-[n+2] = \{\mathbf{z}_j^-[n+2]: j \in J^-\}
 \tag{22}$$

The *positive set* $Z^+[n+2]$ is composed of m^+ augmented vectors $\mathbf{z}_j^+[n+2]$ ($\mathbf{z}_j^+[n+2] \neq \mathbf{0}$) and the *negative set* $Z[n+2]$ is composed of m^- augmented vectors $\mathbf{z}_j^-[n+2]$ ($\mathbf{z}_j^-[n+2] \neq \mathbf{0}$) (18).

Definition 4: The sets $Z^+[n+2]$ and $Z[n+2]$ (22) of the augmented feature vectors $\mathbf{z}_j^+[n+2]$ and $\mathbf{z}_j^-[n+2]$ are *linearly separable*, if and only if there exists such augmented vector of parameters $\mathbf{v}'[n+2]$, that all the inequalities (21) are fulfilled.

The following *Lemma* can be proved:

Lemma 1: All the linear inequalities (10), (11) and (12) can be fulfilled by some parameters vector $\mathbf{v}'[n+2] = [\mathbf{w}'[n]^T, \theta', \beta']$ (19) if and only if the sets $Z^+[n+2]$ and $Z[n+2]$ (22) are linearly separable (*Definition 3*).

4 Convex and Piecewise Linear (CPL) Criterion Function for Survival Data Sets

The *augmented feature vectors* $\mathbf{z}_j^+[n+2]$ and $\mathbf{z}_j^-[n+2]$ (18) and the *augmented weight vector* $\mathbf{v}[n+2]$ (19) have been introduced in the case of the interval regression model []. The positive set $Z^+[n+2]$ and the negative set $Z[n+2]$ (22) are composed of such $(n + 2)$ - dimensional vectors $\mathbf{z}_j^+[n+2]$ and $\mathbf{z}_j^-[n+2]$ (18) which are different from zero. The family of linear inequalities (21) represents the concept of linear separability of the sets $Z^+[n+2]$ and $Z[n+2]$ (22).

The below convex and piecewise-linear (CPL) penalty functions $\phi_j^+(\mathbf{v}[n+2])$ and $\phi_j^-(\mathbf{v}[n+2])$ are introduced for the purpose of the inequalities (10), (11) and (12) reinforcement:

$$\begin{aligned} & (\forall \mathbf{z}_j^+[n+2] \neq \mathbf{0}) \\ & \phi_j^+(\mathbf{v}[n+2]) = \begin{cases} 1 - \mathbf{v}[n+2]^T \mathbf{z}_j^+[n+2] & \text{if } \mathbf{v}[n+2]^T \mathbf{z}_j^+[n+2] < 1 \\ 0 & \text{if } \mathbf{v}[n+2]^T \mathbf{z}_j^+[n+2] \geq 1 \end{cases} \end{aligned} \quad (23)$$

$$\begin{aligned} & (\forall \mathbf{z}_j^-[n+2] \neq \mathbf{0}) \\ & \phi_j^-(\mathbf{v}[n+2]) = \begin{cases} 1 + \mathbf{v}[n+2]^T \mathbf{z}_j^-[n+2] & \text{if } \mathbf{v}[n+2]^T \mathbf{z}_j^-[n+2] > -1 \\ 0 & \text{if } \mathbf{v}[n+2]^T \mathbf{z}_j^-[n+2] \leq -1 \end{cases} \end{aligned} \quad (24)$$

The *perceptron criterion function* $\Phi(\mathbf{v}[n+2])$ is defined as the weighted sum of the penalty functions $\phi_j^+(\mathbf{v}[n+2])$ (23) and $\phi_j^-(\mathbf{v}[n+2])$ (24) []:

$$\Phi(\mathbf{v}[n+2]) = \sum_j \alpha_j \phi_{H_j}^+(\mathbf{v}[n+2]) + \sum_j \alpha_j \phi_{H_j}^-(\mathbf{v}[n+2]) \quad (25)$$

where positive parameters α_j ($\alpha_j \geq 0$) determine an *importance* of the particular vectors $\mathbf{z}_j^+[n+2]$ or $\mathbf{z}_j^-[n+2]$ (18).

The optimal vector $\mathbf{v}^*[n+2]$ constitutes the minimum of the *CPL* criterion function $\Phi(\mathbf{v}[n+2])$ (25):

$$(\forall \mathbf{v}[n+2]) \quad \Phi(\mathbf{v}[n+2]) \geq \Phi(\mathbf{v}^*[n+2]) = \Phi^* \geq 0 \quad (26)$$

where $\mathbf{v}^*[n+2] = [\mathbf{w}^*[n]^T, w_0^*, \beta^*]^T$, and $\mathbf{w}^*[n] = [w_1^*, \dots, w_n^*]^T$ (19).

The below theorem can be proved [9]:

Theorem 1: The minimal value $\Phi^* = \Phi(\mathbf{v}^*[n+2])$ (26) of the non-negative criterion function $\Phi(\mathbf{v}[n+2])$ (25) is equal to zero ($\Phi^* = 0$) and the sets $Z^+[n+2]$ and $Z[n+2]$ (22) are linearly separable (*Definition 3*) if and only if there exists such weight vector $\mathbf{w}^*[n]$ and the threshold θ' , that all the inequalities (10), (11) and (12) based on the survival set C_4 (9) are fulfilled.

Remark 2: If the minimal value $\Phi^* = \Phi(\mathbf{v}^*[n+2])$ (26) is equal to zero ($\Phi^* = 0$) in the point $\mathbf{v}^*[n+2] = [\mathbf{w}^*[n]^T, \theta^*, \beta^*]^T$ with $\beta^* > 0$, then the below optimal prognostic model $T^*(\mathbf{x}[n])$ (6) fulfils all the inequalities (10), (11) and (12).

$$T^*(\mathbf{x}[n]) = (\mathbf{w}^*[n] / \beta^*)^T \mathbf{x}[n] + w_0^* / \beta^* \quad (27)$$

It means, that all the inequalities (10), (11) and (12) are fulfilled:

$$(\forall j \in \{1, \dots, m\}) \quad y_j^- < (\mathbf{w}^*[n] / \beta^*)^T \mathbf{x}_j[n] + w_0^* / \beta^* < y_j^+ \quad (28)$$

where $y_j^- = t_j^-$ or $y_j^- = -\infty$ and $y_j^+ = t_j^+$ or $y_j^+ = +\infty$.

If the minimal value Φ^* (42) is greater than zero ($\Phi^* > 0$) in the point $\mathbf{v}^*[n+2]$, then the optimal model (28) does not fulfil all the inequalities (10), (11) and (12).

5 Selection of Prognostic Feature Subset with the Use of the RLS Method

The *CPL* criterion function $\Phi(\mathbf{v}[n+2])$ (25) can be modified for the purpose of feature selection by inclusion of the *feature penalty functions* $\phi_i(\mathbf{v}[n+2])$ and the *costs* γ_i ($\gamma_i > 0$) related to particular features x_i [11]. The penalty functions $\phi_i(\mathbf{v}[n+1])$ are defined by:

$$(\forall i \in \{1, \dots, n\}) \quad \phi_i(\mathbf{v}[n+2]) = |\mathbf{e}_i[n+1]^T \mathbf{v}[n+2]| = |w_i| \quad (29)$$

The modified criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ is the sum of the *perceptron criterion function* $\Phi(\mathbf{v}[n+2])$ (25) and a regularization component []:

$$\Psi_\lambda(\mathbf{v}[n+2]) = \Phi(\mathbf{v}[n+2]) + \lambda \sum_{i \in \{1, \dots, n\}} \gamma_i \phi_i(\mathbf{v}[n+2]) = \Phi(\mathbf{v}[n+2]) + \lambda \sum_{i \in \{1, \dots, n\}} \gamma_i |w_i| \quad (30)$$

where λ ($\lambda \geq 0$) is the *cost level* ($\lambda \geq 0$).

Standard assumption about the *feature costs* γ_i is such that these costs are equal one:

$$(\forall i \in \{1, \dots, n\}) \quad \gamma_i = 1 \tag{31}$$

The modified criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) is used in the *relaxed linear separability (RLS)* method of feature subset selection []. The regularization component $\lambda \sum \gamma_i |w_i|$ used in the modified criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) is similar to that used in the *Lasso* method []. The *Lasso* method was developed in the framework of the regression analysis for the model selection purposes []. The main difference between the *Lasso* and the *RLS* methods is in the types of the basic criterion functions. The basic criterion function used in the *Lasso* method is the *Last squares* type. The basic criterion function $\Phi(\mathbf{v}[n+2])$ (25) used in the *RLS* method is the *CPL* type. This difference affects, inter alia, the computational techniques used to minimize of the criterion functions.

The criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30), similarly to the function $\Phi(\mathbf{v}[n+2])$ (25) is convex and piecewise-linear (*CPL*). The basis exchange algorithms allow to find efficiently the optimal vector of parameters (*vertex*) $\mathbf{v}_\lambda^*[n+2]$ constituting minimum of the criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) with the cost level λ []:

$$(\exists \mathbf{v}_\lambda^*[n+2]) \quad (\forall \mathbf{v}[n+2]) \quad \Psi_\lambda(\mathbf{v}[n+2]) \geq \Psi_\lambda(\mathbf{v}_\lambda^*[n+2]) = \Psi_\lambda^* \tag{32}$$

The parameters $\mathbf{v}_\lambda^*[n+2] = [\mathbf{w}_\lambda^*[n], \theta_\lambda^*, \beta_\lambda^*]^T = [w_{\lambda 1}^*, \dots, w_{\lambda n}^*, \theta_\lambda^*, \beta_\lambda^*]^T$ (19) define the optimal prognostic model $T^*(\mathbf{x})$ (27).

The below hyperplane $h_j^+[n+2]$ in the parameter space R^{n+2} has been related to each augmented feature vector $\mathbf{z}_j^+[n+2]$ (18) from the set H^+ (22) []. Similarly, the hyperplane $h_j^-[n+2]$ has been related to each augmented feature vector $\mathbf{z}_j^-[n+2]$ (18) from the set H^- (22).

$$\begin{aligned} (\forall j \in J^+) \quad h_j^+[n+2] &= \{\mathbf{v}[n+2]: \mathbf{z}_j^+[n+2]^T \mathbf{v}[n+2] = 1\} \\ &\quad \text{and} \\ (\forall j \in J^-) \quad h_j^-[n+2] &= \{\mathbf{v}[n+2]: \mathbf{z}_j^-[n+2]^T \mathbf{v}[n+2] = -1\} \end{aligned} \tag{33}$$

The first n unit vectors $\mathbf{e}_i[n+2] = [0, \dots, 0, 1, 0, \dots, 0]^T$ ($i = 1, \dots, n$) without the vectors $\mathbf{e}_{n+1}[n+2] = [0, \dots, 0, 1, 0]^T$ and $\mathbf{e}_{n+2}[n+2] = [0, \dots, 0, 1]^T$ are used in defining the hyperplanes $h_i^0[n+2]$ in the augmented parameter space R^{n+2} (19):

$$(\forall i \in \{1, \dots, n\}) \quad h_i^0[n+2] = \{\mathbf{v}[n+2]: \mathbf{e}_i[n+2]^T \mathbf{v}[n+2] = 0\} = \{\mathbf{v}[n+2]: v_i = 0\} \tag{34}$$

The hyperplanes $h_j^+[n+2]$, $h_j^-[n+2]$ and $h_i^0[n+2]$ divide the parameter space R^{n+2} (19) in the disjointed regions $R_k[n+2]$. Each region $R_k[n+2]$ is a convex polyhedron in the parameter space with number of vertices $\mathbf{v}_k[n+2]$. The *CPL* criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) is linear inside each region $R_k[n+2]$. It has been shown based on the theory of linear programming that the minimum of the *CPL* criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) can be found in one of vertices $\mathbf{v}_k[n+2]$ of some region $R_k[n+2]$ []. Each vertex $\mathbf{v}_k[n+2]$ in the parameter space R^{n+2} is the intersection point of at least $(n + 2)$ hyperplanes $h_j^+[n+2]$, $h_j^-[n+2]$ (34) or $h_i^0[n+2]$ (33). The below equations are fulfilled in each vertex $\mathbf{v}_k[n+2]$:

$$(\forall j \in J_k^+) \quad \mathbf{z}_j^+[n+2]^T \mathbf{v}_k[n+2] = 1, \text{ and}$$

$$\begin{aligned} (\forall j \in J_k^-) \quad \mathbf{z}_j^-[n+2]^T \mathbf{v}_k[n+2] &= -1, \text{ and} \\ (\forall i \in I_k^0) \quad \mathbf{e}_i[n+2]^T \mathbf{v}_k[n+2] &= 0 \end{aligned} \quad (35)$$

where J_k^+ and J_k^- are the sets of indices j such hyperplanes $h_j^+[n+2]$, $h_j^-[n+2]$ (34) that pass through the vertex $\mathbf{v}_k[n+2]$, I_k^0 is the set of indices i such hyperplanes $h_i^0[n+2]$ (34) that pass through the vertex $\mathbf{v}_k[n+2]$.

The above equations can be given in the matrix form:

$$\mathbf{B}_k[n+2] \mathbf{v}_k[n+2] = \boldsymbol{\delta}_k'[n+2] \quad (36)$$

where $\mathbf{B}_k[n+2]$ is the non-singular matrix (*basis*) with the rows constituted by the linearly independent vectors $\mathbf{z}_j^+[n+2]$ ($j \in J_k^+$), $\mathbf{z}_j^-[n+2]$ ($j \in J_k^-$) (18) or the unit vectors $\mathbf{e}_i[n+2]$ ($i \in I_k^0$), and $\boldsymbol{\delta}_k'[n+2]$ is the *margin vector* with components equal to 1, -1 or 0 according to (35).

Remark 3: Such features x_i which are linked to the unit vectors $\mathbf{e}_i[n+2]$ ($i \in I_k^0$) in the basis $\mathbf{B}_k[n+2]$ (37) have the weights w_i equal to zero ($w_i = 0$) in the vector $\mathbf{v}_k[n+1] = [\mathbf{w}_k[n]^T, \theta_k, \beta_k]^T = [w_1, \dots, w_n, \theta_k, \beta_k]^T$.

This *Remark* can be justified by the below implication (35):

$$(\forall i \in I_k^0) \quad (\mathbf{e}_i[n+2]^T \mathbf{v}_k[n+2] = 0) \Rightarrow (w_i = 0) \quad (37)$$

We can remark, that such features x_i which have the weights $w_{\lambda_i}^*$ equal to zero ($w_{\lambda_i}^* = 0$) in the optimal vector $\mathbf{v}_{\lambda}^*[n+2]$ (32) can be reduced without changing the optimal prognostic model $T^*(\mathbf{x})$ (27) due to feature vectors $\mathbf{x}[n]$. In consequence, the feature reduction rule has been based on the weights $w_{\lambda_i}^*$ equal to zero []:

$$(w_{\lambda_i}^* = 0) \Rightarrow (\text{the feature } x_i \text{ is reduced}) \quad (38)$$

The minimal value $\Psi_{\lambda}(\mathbf{v}_{\lambda}^*[n+2])$ (32) of the *CPL* criterion function $\Psi_{\lambda}(\mathbf{v}[n+2])$ (30) in the point $\mathbf{v}_{\lambda}^*[n+2]$ represents an equilibrium between a "force" of linear dependency (6) and features costs determined by the parameters λ and γ_i . The optimal vertex $\mathbf{v}_k^*[n+2] = \mathbf{v}_{\lambda}^*[n+2]$ (32) is linked to the optimal basis $\mathbf{B}_k^*[n+2]$ by the equation (36). An increase of the parameter λ value in the criterion function $\Psi_{\lambda}(\mathbf{v}[n+2])$ (30) results in an increase of the number of the unit vectors in the optimal basis $\mathbf{B}_k^*[n+2]$ (36). We can remark that an increase of the *cost level* λ value in the minimized function $\Psi_{\lambda}(\mathbf{v}[n+2])$ (30) can result in an increased number of the reduced features x_i (33). In consequence, the dimensionality of the feature $F[n]$ can be reduced arbitrarily by a sufficient increase of the parameter λ in the criterion function $\Psi_{\lambda}(\mathbf{v}[n+2])$ (30). Such method of feature selection has been named *relaxed linear separability (RLS)* [11]. A successive increase of the *cost level* λ in the minimized function $\Psi_{\lambda}(\mathbf{v}[n+2])$ (30) allows to reduce a less important (redundant) features x_i and to generate the descending sequence of feature subspaces $F_k[n_k]$ ($n_k > n_{k+1}$):

$$F[n] \supset F_1[n_1] \supset \dots \supset F_k[n_k], \text{ where } 0 \leq \lambda_0 < \lambda_1 < \dots < \lambda_k \quad (39)$$

Each feature subspace $F_k[n_k]$ in the above sequence has been linked to a certain value λ_k of the cost level λ in the criterion function $\Psi_{\lambda}(\mathbf{v}[n+2])$ (30). The sequence (39) of

the feature subspaces $F_k[n_k]$ can be generated in a deterministic manner on the basis of the learning set C_4 (9) in accordance with the *relaxed linear separability (RLS)* method [1]. Each step $F_k[n_k] \rightarrow F_{k+1}[n_{k+1}]$ can be realized by a minimal increase $\lambda_k \rightarrow \lambda_{k+1} = \lambda_k + \Delta_k$ (where $\Delta_k > 0$) of the cost level λ in the criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30).

A high value λ_k of the cost level λ in criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) can make all vectors $\mathbf{z}_j^+[n+2]$ or $\mathbf{z}_j^-[n+2]$ (18) in the optimal basis $\mathbf{B}_k^*[n+2]$ (36) will be replaced by unit vectors $\mathbf{e}_i[n+2]$ and $\mathbf{w}_k^*[n] = 0$. This would mean that all features x_i are eliminated (38), which is not a constructive solution. A compromise solution is needed which allow to preserve the most important feature subset. Such postulate can be realized through an adequate stop criterion in the process of the feature space $F[n]$ reduction (39). Such stop criterion should be based on evaluation of feature subspaces $F_k[n_k]$ (39) quality.

In accordance with the *relaxed linear separability (RLS)* approach to feature subset selection, a quality of a given subspace $F_k[n_k]$ (39) is evaluated on the basis of the optimal linear classifier designed in this subspace [11]. The better optimal linear classifier means the better feature subspace $F_k[n_k]$.

Similar approach can be applied also for the purpose of the prognostic model $T(\mathbf{x})$ (6) selection. The optimal linear classifier is defined in the feature subspace $F_k[n_k]$ by the below decision rule:

$$\begin{aligned} \text{if } \mathbf{v}^*[n_k]^T \mathbf{y}[n_k] \geq 0, \text{ then } \mathbf{z}[n_k] \text{ is allocated to the to the set } H_k^+ \\ \text{if } \mathbf{v}^*[n_k]^T \mathbf{y}[n_k] < 0, \text{ then } \mathbf{z}[n_k] \text{ is allocated to the to the set } H_k^- \end{aligned} \quad (40)$$

where the optimal vector $\mathbf{v}_k^*[n_k]$ constitutes the minimum (26) of the *CPL* criterion function $\Phi_k(\mathbf{v}[n_k])$ (25). The criterion function $\Phi_k(\mathbf{v}[n_k])$ is defined (25) on the reduced feature vectors $\mathbf{z}_i^+[n_k]$ and $\mathbf{z}_i^-[n_k]$ (18) belonging to the feature subspace $F_k[n_k]$ (40). The vectors $\mathbf{z}_i^+[n_k]$ and $\mathbf{z}_i^-[n_k]$ are obtained from the feature vectors $\mathbf{z}_i^+[n+2]$ and $\mathbf{z}_j^-[n+2]$ (18) by reducing some features x_i in accordance with the sequence (39). The sets H_k^+ and H_k^- contains the vectors $\mathbf{z}_i^+[n_k]$ and $\mathbf{z}_i^-[n_k]$ according to (22).

The quality of the linear classifiers (40) can be evaluated by using the error estimator (*apparent error rate*) $e_a(\mathbf{v}^*[n_k])$ as the fraction of wrongly classified elements $\mathbf{z}_j^+[n_k]$ and $\mathbf{z}_j^-[n_k]$ (18) from the sets H_k^+ and H_k^- (22):

$$e_a(\mathbf{v}^*[n_k]) = m_a(\mathbf{v}^*[n_k]) / m \quad (41)$$

where m is the number of all elements $\mathbf{z}_j^+[n_k]$ and $\mathbf{z}_j^-[n_k]$ (18) from the sets H_k^+ and H_k^- (22) and $m_a(\mathbf{v}^*[n_k])$ is the number of such elements which are wrongly allocated by the rule (40).

It is known that if the same vectors $\mathbf{x}_j[n_k]$ are used for classifier designing and classifier evaluation, then the evaluation results are too optimistic (*biased*) [1]. For the purpose of the bias reduction of the apparent error rate estimator $e_a(\mathbf{v}^*[n_k])$ (41), the cross-validation error rate $e_{CV\bar{E}}(\mathbf{v}^*[n_k])$ (41) is evaluated [1].

In accordance with the *RLS* method, a successive feature subspaces $F_k[n_k]$ in the descending sequence (39) are evaluated by using the cross-validation error rate

$e_{\text{CVE}}(\mathbf{v}^*[n_k])$ (41). It was assumed, that the optimal feature subspace $F_k[n_k]$ (39) is characterised by the lowest error rate $e_{\text{CVE}}(\mathbf{v}^*[n_k])$ (41).

An example of another stop criterion in the feature reduction (39) can be found in the work [INTECH]. In this example, the feature reduction in the genetic data sets were carried out until the linear separability of these sets were preserved. Such approach is particularly useful in the case of genetic data sets, when the number of objects is low in comparison to the number of features (*genes*) which have been used to characterise these objects.

In the case of the prognostic model $T(\mathbf{x})$ (6) selection, a quality of a given subspace $F_k[n_k]$ (39) can be evaluated also on the basis of the optimal linear prognostic model $T^*(\mathbf{x})$ (27) designed in this subspace. The better prognostic model $T^*(\mathbf{x})$ (27) means the better feature subspace $F_k[n_k]$.

6 Example 1: Prognostic Model Selection on High Dimensional Synthetic Data Set

The synthetic data set contained $m = 1000$ feature vectors $\mathbf{x}_j[n]$, where $n = 100$ []. The vectors $\mathbf{x}_j[n]$ were generated randomly in accordance with the multivariate normal distribution $N(\mathbf{0}, \mathbf{\Sigma})$, where the covariance matrix $\mathbf{\Sigma}$ is equal to the unit matrix \mathbf{I} ($\mathbf{\Sigma} = \mathbf{I}$). The prognostic model $y(\mathbf{x}[n])$ (1) were defined a priori as the below linear combination of 10 features x_i (the *linear key*):

$$y(\mathbf{x}[n]) = 3x_3 + 4x_9 - 7x_{16} + 2x_{27} - 6x_{35} + 3x_{40} + 3x_{57} - 8x_{62} + x_{74} - x_{91} + 5 \quad (42)$$

The learning set C'_1 (2) was randomly generated this way:

$$C'_1[n] = \{\mathbf{x}_j[n]; y(\mathbf{x}_j[n])\}, \text{ where } j = 1, \dots, m \quad (43)$$

The set $C'_1[n]$ was replaced by the interval learning set $C_{\varepsilon 2}'[n]$ (5):

$$C_{\varepsilon 2}'[n] = \{\mathbf{x}_j[n], [y(\mathbf{x}_j[n]) - \varepsilon, y(\mathbf{x}_j[n]) + \varepsilon]\} \quad (44)$$

where ε ($\varepsilon > 0$) is the *margin* (Remark 1) and $j = 1, \dots, m$.

On the basis of the interval learning set $C_{\varepsilon 2}'[n]$ (44) the differentiated data sets $Z_{\varepsilon}^+[n+2]$ and $Z_{\varepsilon}^-[n+2]$ (22) has been created. The modified criterion function $\Psi_{\lambda}(\mathbf{v}[n+2])$ (30) has been defined on the elements $\mathbf{z}_{\varepsilon j}^+[n+2]$ of the set $H_{\varepsilon}^+[n+2]$ and on the elements $\mathbf{z}_{\varepsilon j}^-[n+2]$ of the set $H_{\varepsilon}^-[n+2]$ (22), where (18):

$$\begin{aligned} \mathbf{z}_{\varepsilon j}^+[n+2] &= [\mathbf{x}_j[n]^T, 1, -y(\mathbf{x}_j[n]) + \varepsilon]^T, \text{ and} \\ \mathbf{z}_{\varepsilon j}^-[n+2] &= [\mathbf{x}_j[n]^T, 1, -y(\mathbf{x}_j[n]) - \varepsilon]^T \end{aligned} \quad (45)$$

The parameters $\mathbf{v}_{\varepsilon \lambda}^*[n+2] = [\mathbf{w}_{\varepsilon \lambda}^*[n]^T, \theta_{\varepsilon \lambda}^*, \beta_{\varepsilon \lambda}^*]^T = [w_{\lambda 1}^*, \dots, w_{\lambda n}^*, \theta_{\lambda}^*, \beta_{\lambda}^*]^T$ (19) define the minimal value of the criterion function $\Psi_{\varepsilon \lambda}(\mathbf{v}[n+2])$ (30) with the cost level λ and the margin ε ($\varepsilon > 0$):

$$\mathbf{v}_{\varepsilon \lambda}^*[n+2] = \text{argmin } \Psi_{\varepsilon \lambda}(\mathbf{v}[n+2]) \quad (46)$$

The parameters $\mathbf{v}_{\varepsilon\lambda}^*[n+2]$ define also the optimal prognostic model $T_{\varepsilon\lambda}^*(\mathbf{x}[n])$ (27).

Using synthetic data set two sets of experiments were performed. In the first one the interval learning set $C_{\varepsilon 2}'[n]$ (44) was generated with using different values of the parameter ε . Experiments were designed to check if the ε margin affect the ability of the algorithm to rediscover the key (42) encoded in data. The following values for ε were used: 0.001, 0.01, 0.1, 0.5, 1, 2, 3, 5. It appeared that only in case of the largest values of ε ($\varepsilon = 3$ and $\varepsilon = 5$), the key has not been rediscovered in its entirety. In other cases, the encoded sequence was exactly rediscovered.

The second set of experiments was to test the impact of the presence of censored observations in the data set on the ability of the algorithm to find the encoded key (43). The simulation of censoring was achieved by rejecting a certain number of elements $\mathbf{z}_{\varepsilon_j^+}[n+2]$ and $\mathbf{z}_{\varepsilon_j^-}[n+2]$ (45) from the differentiated data sets $Z_{\varepsilon^+}[n+2]$ and $Z_{\varepsilon^-}[n+2]$ (22). For each pair $\mathbf{z}_{\varepsilon_j^+}[n+2]$, $\mathbf{z}_{\varepsilon_j^-}[n+2]$ ($j=1, \dots, 1000$) (45) were drawn with a fixed probability p rejection of one of the elements of pair. If the lottery were positive, with a probability equal to 0.5 was rejected element $\mathbf{z}_{\varepsilon_j^+}[n+2]$ (right censoring) or $\mathbf{z}_{\varepsilon_j^-}[n+2]$ (left censoring). Calculations were performed using the values of probability p from 0.1 to 1 (step 0.1). In any case, coded key (42) was correctly rediscovered.

The *RLS* method allowed to reduce 90 features x_i while preserving the linear separability of the reduced learning sets $G_k^+[10]$ and $G_k^-[10]$. The linear key (42) of 10 features x_i has been rediscovered by the *RLS* method [2].

7 Example 2: Prognostic Model Selection on the Breast Cancer Survival Data Set

The Breast Cancer dataset consists of patient samples from primary invasive breast carcinomas. The dataset collected by van't Veer et al. [12] was comprised of 97 objects. Previously van't Veer et al. identified a 70-gene predictive signature which classified objects into good and poor prognosis groups. Subsequently, van de Vijver et al. [13] acquired a test set of 295 objects with clinical data on which to validate the 70-gene predictive signature. The objects in both breast cancer datasets contains approximately 25000 genes. Yeung et al. [14] filtered the dataset down to 4919 significantly regulated genes (at least a 2-fold difference and p-value < 0.01 in at least three objects), and we have chosen to conduct our analysis with these 4919 genes. Each object has a specified time value measured from start of observation until death or censoring. 216 patients (73%) were still alive at the final follow-up visit (censoring observations).

The aim of the experiment was to determine the predictive model $T^*(\mathbf{x}[n])$ (27) allowing to calculate the patient expected survival time on the base on the value of selected attributes.

On the basis of 295 objects from the Breast cancer data set 374 elements $\mathbf{z}_j^+[n+2]$ and $\mathbf{z}_j^-[n+2]$ (18) were created. Then the *RLS* method was applied to the newly formed data set.

According to the first stop criterion of the *RLS* method, the smallest value of cross-validation error rate $e_{\text{CVE}}(\mathbf{v}^*[n_k])$ (41) 1.337% has been reached in the feature space

constructed from 99 features. However linear separability of elements $\mathbf{z}_j^+[n+2]$ and $\mathbf{z}_j^-[n+2]$ (18) has been preserved even in the feature space of size 58. In this case cross-validation error rate $e_{\text{CVE}}(\mathbf{v}^*[n_k])$ (41) was equal to 15.775%.

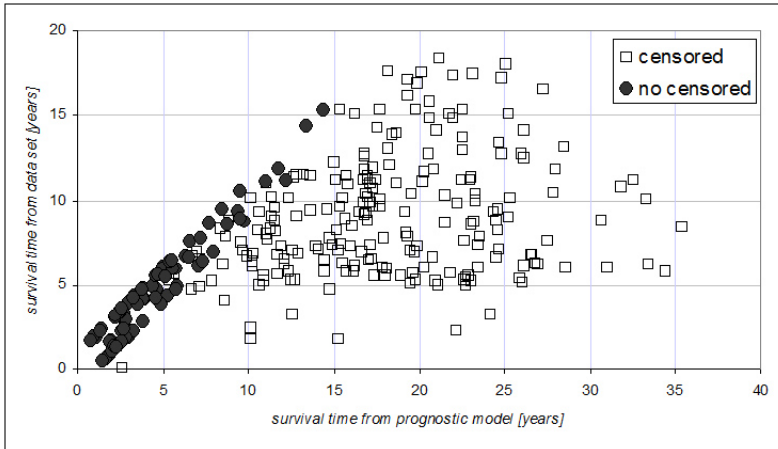


Fig. 1. The values of survival time calculated on the basis of the model $T^*(\mathbf{x}[58])$ compared with the values of survival time given in *Breast cancer* data set

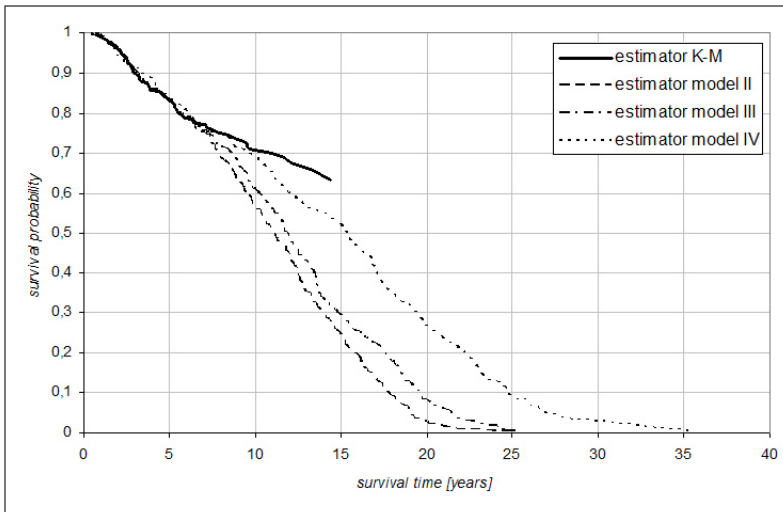


Fig. 2. Survival probability: K-M – estimator Kaplana-Meier’a, model II - $T^*(\mathbf{x}[160])$ maximal margin, model III - $T^*(\mathbf{x}[99])$ minimal CVE, model IV - $T^*(\mathbf{x}[58])$ second RLS stop criterion

Figure 1 presents survival times calculated on the basis of the obtained model $T^*(\mathbf{x}[58])$ compared to survival times from the Breast cancer data set. In case of no censored observations, where the survival time specified in the data set is the real

survival time, values of time calculated from the model are very close to the values saved in the data set. In the case of censored observations, where the survival time saved in the data set is the lower limit of real survival time, values of time calculated from the model are in any case greater than the corresponding time values from the data set.

8 Concluding Remarks

The task of linear regression model designing has been reformulated here and solved as a problem of testing the linear separability of two sets. In result, the problem of linear regression model designing has been replaced by the problem of linear classifier designing. Such reformulation allowed to tread in the same manner multivariate learning sets with different types of dependent variables. The learning sets with right, left or interval type censored dependent variables have been treated in the same manner as the learning sets with uncensored dependent variables.

The problem of linear classifier designing has been solved through minimization of the convex and piecewise linear (CPL) criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30). The basis exchange algorithms allow to find efficiently the optimal vector of parameters (*vertex*) $\mathbf{v}_\lambda^*[n+2]$ constituting minimum of the criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) with the cost level λ []. The relaxed linear separability (RLS) method of feature subset selection has been applied to multivariate learning sets with censored dependent variables []. The RLS method involved multiple minimization of the criterion function $\Psi_\lambda(\mathbf{v}[n+2])$ (30) with different values of the cost level λ [].

The proposed method of the linear regression models designing has been tested both on synthetic data set [] as well as on genetic data set with censored survival times []. The synthetic data set contained the linear regression model $y(\mathbf{x}[n])$ (42) based on 10 variables x_i , which was hidden in the random values of 90 other variables x_i . The RLS method allowed to find the hidden model (42) in the set containing $m = 1000$ feature vectors $\mathbf{x}_i[n]$, with $n = 100$. The model $y(\mathbf{x}[n])$ (42) has been found even in the case, where all values y_j of the dependent variable y were censored.

The experiments were carried out also on the genetic data set *Brest cancer* []. These experiments demonstrated, inter alia, that the RLS method allows to find not too numerous (=) subsets of genes x_i with interesting properties, even if the number of genes x_i at the beginning is a huge (=). The found subsets of genes allowed to design reasonable prognostic models (Fig, ?).

Acknowledgment. This work was supported by the by the NCBiR project N R13 0014 04, and partially financed by the project S/WI/2/2012 from the Białystok University of Technology, and by the project 16/St/2012 from the Institute of Biocybernetics and Biomedical Engineering PAS.

References

1. Johnson, R.A., Wichern, D.W.: Applied Multivariate Statistical Analysis. Prentice-Hall, Inc., Englewood Cliffs (1991)
2. Duda, O.R., Hart, P.E., Stork, D.G.: Pattern Classification. J. Wiley, New York (2001)

3. Bobrowski, L.: Ranked linear models and sequential patterns recognition. *Pattern Analysis & Applications* 12(1), 1–7 (2009)
4. Buckley, J., James, I.: Linear regression with censored data. *Biometrika* 66, 429–436 (1979)
5. Gomez, G., Espinal, A., Lagakos, S.: Inference for a linear regression model with an interval-censored covariate. *Statistics in Medicine* 22, 409–425 (2003)
6. Klein, J.P., Moeschberger, M.L.: *Survival Analysis. Techniques for Censored and Truncated Data*. Springer, NY (1997)
7. Bobrowski, L.: Liniowe modele prognostyczne oparte na regresji przedziałowej z funkcjami typu *CPL* (in Polish) Linear prognostic models based on interval regression with CPL functions. *Symulacja w Badaniach i Rozwoju* 1, 109–117 (2010)
8. Bobrowski, L.: Selection of High Risk Patients with Ranked Models Based on the CPL Criterion Functions. In: Perner, P. (ed.) *ICDM 2010. LNCS*, vol. 6171, pp. 432–441. Springer, Heidelberg (2010)
9. Bobrowski, L.: Eksploracja danych oparta na wypukłych i odcinkowo-liniowych funkcjach kryterialnych (in Polish) Data Mining Based on Convex and Piecewise Linear Criterion Functions. Technical University Białystok (2005)
10. Bobrowski, L.: Design of piecewise linear classifiers from formal neurons by some basis exchange technique. *Pattern Recognition* 24(9), 863–870 (1991)
11. Bobrowski, L., Łukaszuk, T.: Feature selection based on relaxed linear separability. *Bio-cybernetics and Biomedical Engineering* 29(2), 43–59 (2009)
12. van 't Veer, L.J., Dai, H., van de Vijver, M.J., He, Y.D., Hart, A.A., Mao, M., Peterse, H.L., van der Kooy, K., Marton, M.J., Witteveen, A.T., Schreiber, G.J., Kerkhoven, R.M., Roberts, C., Linsley, P.S., Bernards, R., Friend, S.H.: Gene expression profiling predicts clinical outcome of breast cancer. *Nature* 415, 530–536 (2002)
13. van der Vijver, M.J., He, Y.D., van 't Veer, L.J., Dai, H., Hart, A.A., Voskuil, D.W., Schreiber, G.J., Peterse, J.L., Roberts, C., Marton, M.J., Parrish, M., Atsma, D., Witteveen, A., Glas, A., Delahaye, L., van der Velde, T., Bartelink, H., Rodenhuis, S., Rutgers, E.T., Friend, S.H., Bernards, R.: A gene-expression signature as a predictor of survival in breast cancer. *N. Engl. J. Med.* 347, 1999–2009 (2002)
14. Yeung, K., Bumgarner, R., Raftery, A.E.: Bayesian Model Averaging: Development of an Improved Multi-Class, Gene Selection and Classification Tool for Microarray Data. *Bioinformatics* 21, 2394–2402 (2005)

SHACUN: Semi-supervised Hierarchical Active Clustering Based on Ranking Constraints

Eya Ben Ahmed, Ahlem Nabli, and Faïez Gargouri

University of Sfax
eya.benahmed@gmail.com,
{ahlem.nabli,faiez.gargouri}@fsegs.rnu.tn

Abstract. Semi-supervised approaches have proven to be efficient in clustering tasks. They allow user input, thus enhancing the quality of the clustering. However, the user intervention is generally limited to integrate boolean constraints in form of *must-link* and *cannot-link* constraints between pairs of objects. This paper investigates the issue of satisfying ranked constraints in performing hierarchical clustering. *SHACUN* is a new introduced method for handling cases when some constraints are more important than others and must be firstly enforced. Carried out experiments on real log files used for decision-maker groupization in data warehouse confirm the soundness of our approach.

Keywords: semi-supervised clustering, hierarchical clustering, ranking constraints, groupization, data warehouse.

1 Introduction

Clustering techniques [10] are among the outstanding machine learning techniques to discover groups and identify distinguished clusters in the considered data. These techniques are used in several domains such as finance, stock market, banking, etc. They work under an unsupervised mode when the class label of each object in the data set is not known a priori. Nevertheless, provided results by clustering techniques, in most real situations, do not always fit the expert expectations due to the unsupervised aspect of such methods. In order to overcome this drawback, many researches have been done to integrate external knowledge in the clustering process [5]. The external knowledge is conveyed in form of constraints. These constraints may be straightforwardly derived from the original data using partially labeled data or provided by expert. Let us note that several approaches have already dealt with the problem of semi-supervised hierarchical clustering. Most of them integrate boolean constraints in form of *must-link* and *cannot-link* constraints between pairs of objects [12].

However, in many real-world situations, expert may prefer the merge of particular objects before others, because sometimes there are multiple possible groupings. Motivated by the issue that produced clusters must be the ones required and suffering from insufficiency of labeled data to apply classification, the main

trust of this paper is to rank incorporated constraints according to their importance. To address the problem caused by the constraint priorities, we propose the ranking of constraints that may be integrated in the clustering process. Two alternatives are possible: (i) we deal with qualitative approach of constraints ranking which aims at a relative formulation of constraints ranking, such as the user prefers "Constraint X" over "Constraint Y". Such a formulation is natural for a human and results; (ii) we focus on quantitative approach of constraints ranking. Constraints ranking is specified indirectly using scoring functions that associate a numeric score with every constraint.

In this present study, we suppose that ranking constraints is not hard constraint, the ranking is not necessarily numerical and does not imply total order. We develop the qualitative ranking of constraints that must be integrated in the hierarchical semi-supervised clustering process. So, we suggest to define our constraints ranking model in order to enhance clustering results.

The remainder of this paper is organized as follows : Section 2 motives our approach through a financial example. Section 3 sketches a thorough study of the related work to the hierarchical semi-supervised clustering. In Section 4, we describe our constraints ranking model. Section 5 introduces our algorithm *SHACUN* for clustering derivation under ranked constraints. Experimental results carried out on real financial data warehouse showing the soundness of our approach are presented in section 6. Finally, Section 7 gives a conclusion and future research directions.

2 Motivating Example

To motivate our contribution, we use, throughout this paper, an example of stock exchange data warehouse that we built in order to experiment our contribution. A part of the involved of stock exchange data warehouse is shown in Figure 1. The adopted notations are similar to notations of [8]. This data warehouse aims to analyze the stock average and assess the performance of the stock market. Indeed, our schema contains a fact named *Stock Market Speculation* measured through the *stock exchange index* analyzed according to several dimensions (*Listed company, Time, Title*). Analysis on such a data warehouse may evaluate the performance of equity market and bond markets through the analysis of stock market index of every listed company in a particular sector, either through the measure of sector-based stock index or through the measure of total stock indexes in all sectors.

Such stock exchange data warehouse may orient several decision-makers such as the portfolio managers, the investors, the private administrators and the private investors. In fact, many listed companies are integrated in our data warehouse that's why huge number of analysts may query the built data warehouse. The process of adapting the result of launched query to individual preferences of each analyst seems an effortful task, especially if we consider an analytical history of enough dated navigation [3]. However, such analysts may share analogous interests and have similar preferences, for example, all portfolio managers may

compare the stock market index of their portfolio over the sector-based stock market [2]. An innovative solution is to cluster analysts in groups.

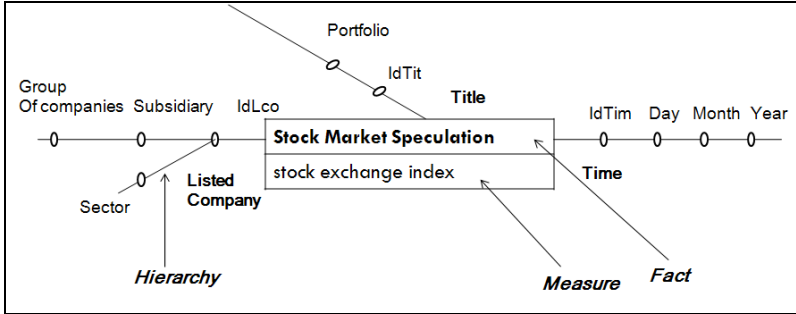


Fig. 1. Stock market data warehouse schema

Indeed, this groupization may be performed in respect to several criteria, namely: (i) the *function exerted*: we assume that analysts working in the same position have similar preferences, (ii) the *granted responsibilities* to accomplish defined goals: in fact two portfolio managers can not assume the same responsibilities, (iii) the *source of group identification*: it is explicit when an analyst specifies to which group he belongs otherwise this task may implicitly performed, (iv) the *dynamic identification* of groups: the detected groups will be updated or remained static.

Hence, we need to supervise the learning process through constraints specification related to identified criteria. The applied constraints are numerous, and may be countless, in respect of studied data warehouse context. Nevertheless, the number of iterations, in some real problems, is greater than the number of required constraints in clustering process. Assuming that some of constraints are more important than others, the ranking of constraints may solve this issue. Accordingly, enforcing ranked constraints during the clustering process, may fit the expert expectations and improve the quality of generated results.

3 Related Work

In this section, we present the related work. Clustering aims to organize a collection of data items into clusters, such that items within a cluster are more "similar" to each other than they are to items in the other clusters [15]. The semi-supervised versions try to improve clustering results by employing external knowledge in the clustering process. The external knowledge is conveyed in form of constraints. These constraints can be directly derived from the original data (using partially labeled data) or provided by user, trying to adapt the clustering results to his/her expectations.

In the sequel, we put the focus on hybrid semi-supervised clustering approaches, particularly those relying on complete-link strategy (see Jain and

Dubes [10]). The main idea behind the latter is the consideration of the distance between one cluster and another cluster to be equal to the longest distance from any member of one cluster to any member of the other cluster. First, Klein et al. [12] used instance-level pairwise constraints (*Must-link* and *Cannot-link*) in a semi-supervised clustering algorithm based on the complete-link algorithm. Constraint insertion has two phases: (i) *Imposition when constraints are integrated to pairs of examples*: The algorithm changes the distance between elements according to required constraints. If two points x_i and x_j have a *Must-link* constraint, then their distance is set to zero. Otherwise, if they have a *Cannot-link* constraint, their distance is set to the maximum distance on the distance matrix plus one. (ii) *Propagation* when the algorithm considers that if an example x_k is near an example x_i and x_i has a *Must-link* or a *Cannot-link* constraint with x_j , so x_k is also near or far from x_j . The new distance between x_k and x_j is calculated through a triangle inequality. Kestler et al. use pairwise constraints on the first level of hierarchical clustering algorithm when generating the initial clusters [11]. Such constraints are not propagated to the posterior levels. Labeled examples are used by Bade et al. in post-processing step [1]. The method used to generate labeled instances *Must-link* constraints and *Cannot-link* between pairs of objects. After a process of unsupervised clustering, these constraints are used to determine whether to merge or split the resulting clusters. Bohm and Plant [4] introduced HISSCLU a hierarchical, density based method for semi-supervised clustering. Instead of deriving explicit constraints from the labeled objects, HISSCLU expands the clusters starting at all labeled objects simultaneously. During the expansion, class labels are assigned to the unlabeled objects most consistently with the cluster structure. Davidson and Ravi presented an agglomerative hierarchical clustering using constraints and demonstrated the enhancement in the clustering accuracy [6]. Their method stops if no more agglomerations according to the *cannot-links* can be performed. The authors prove that the combination of *must-link* and *cannot-link* is computationally viable in a hierarchical clustering unlike flat clustering methods. Nogueira et al. introduce a new active semi-supervised hierarchical clustering method [13]. This strategy uses not only cluster-level constraints [9] where the user can indicate a pair of clusters to be merged but also an innovative concept called *confidence*. When there is lower confidence in a cluster merge the user can be queried and provide a cluster-level constraint.

Exclusively, Klein et al. introduced the only active learning approach where constraints are built in pairs. Restarting the clustering is performed in a non-supervised until the merge step. Then, the user specifies whether the roots of the next merger is supposed to be merged. Depending on the response, the constraints are propagated.

The overview of the main related work leads to deduce that most approaches are based on complete-link algorithm. To the best of our knowledge, as depicted by figure 2, no method has been proposed to apply ordinal ranking of constraints in clustering objects. However, not all required constraints are of high importance. To overcome this drawback, we introduce our new algorithm which applied

ranked constraints in all levels of clustering process. Our major contribution is the injection of such partial order of constraints with respect to their significance to improve the quality of generated clusters. The main thrust of this paper is to propose a new active method for hierarchical clustering relying on ordinal constraints. Indeed, given existing knowledge, we aim to access to unlabelled data, to obtain expensive label then to find labels that are "informative".

Method	Type of learning		Type of linkage			Type of constraint		Step of constraint enforcing	
	Active	Passive	Simple	Average	Complete	Boolean	Ordinal	First level	All levels
(Klein et al., 2002)	x				x	x			x
(Kestler et al., 2006)		x			x	x		x	
(Bade et al., 2007)		x			x	x			x
(Bohm and Plant, 2008)		x			x	x		x	
(Davidson and Ravi, 2009)		x	x		x	x		x	
(Nogueira et al., 2012)		x		x		x		x	
(Our approach, 2012)	x				x		x	x	

Fig. 2. Comparison of semi-supervised hierarchical clustering approaches

4 Constraints Ranking Model

In this section, we introduce our innovative basic concepts that will be of use in the remainder.

4.1 Constraints

Constraints indicate a relationship between two clusters.

Definition 1. Constraint cr

Given two clusters C_i and C_j belonging to the set C of clusters, a constraint cr is a relationship between C_i and C_j . We denote it as follows $cr = (C_i, C_j)$.

Example 1. A typical example of cr constraint is merging the portfolio manager X assigned to the first cluster C_1 and the portfolio manager Y assigned to the second cluster C_2 . Such a constraint is denoted as follows $cr = (C_1, C_2)$.

4.2 Formal Semantic Of Strict Partial Order Of Constraints

A careful examination of ranking reveals that it shares a fundamental common principle. In this familiar setting, it turns out that people express their ranking frequently in terms like "I consider that A is more important than B ". This kind of ranking modeling is widely applied and intuitively understood by everybody.

Definition 2. Ranking $\mathcal{R} = (CR, >\mathcal{R})$

Given a set CR of constraints, a ranking R is a strict partial order $\mathcal{R} = (CR, >\mathcal{R})$, where $>\mathcal{R} \subseteq dom(CR) \times dom(CR)$.

Typical properties of the relation $>\mathcal{R}$ include:

- **Irreflexivity:** $\forall cr_1; cr_1 \not>\mathcal{R} cr_1;$
- **Asymmetry:** $\forall cr_1, cr_2; cr_1 >\mathcal{R} cr_2 \Rightarrow cr_1 \not>\mathcal{R} cr_2;$
- **Transitivity:** $\forall cr_1, cr_2, cr_3; (cr_1 >\mathcal{R} cr_2 \wedge cr_2 >\mathcal{R} cr_3) \Rightarrow cr_1 >\mathcal{R} cr_3;$

The relation $>\mathcal{R}$ is a strict partial order if it is irreflexive, asymmetric and transitive. At this point, we do not assume any properties of $>\mathcal{R}$, although in most applications, it will be at least a strict partial order.

Example 2. Let us consider two constraints cr_1 expressing the relation between two portfolio managers assigned to two clusters C_1 and C_2 and cr_2 expressing relation between the first portfolio manager assigned to the first cluster C_1 and the investor assigned to the cluster C_3 . Those facts may be denoted as follows: $cr_1 = (C_1, C_2)$; and $cr_2 = (C_1, C_3)$. Assuming that it is more important to merge two analysts working in the same function, we infer this knowledge as follows : $cr_1 >\mathcal{R} cr_2$.

4.3 Constraints Ranking Engineering

Complex constraints are abundant, especially in the groupization of decision-makers. Thus, there is a high demand for a powerful and orthogonal framework that supports the complex constraints. We present an inductive approach towards constructing complex rankings. This model will be the key towards a systematic ranking engineering.

A. Non-numerical Base Constraints

We present two fundamental non numerical base constraints, namely *Lowest* and *Highest* constraint.

Definition 3. LOWEST($>\mathcal{R}$)

cr_1 is called the *LOWEST* constraint, if $\forall cr_i \in CR, cr_i >\mathcal{R} cr_1$.

Example 3. In our running example, let us cr_1 be gathering analysts working in the same sector. cr_1 is the lowest constraint compared to other constraints denoted cr_i , such as gathering decision-makers who explicitly choose to belong to given group of analysts or decision-makers working at the same position or assuming the same responsibilities.

Definition 4. HIGHEST($>\mathcal{R}$)

cr_1 is called the *HIGHEST* constraint, if $\forall cr_i \in CR, cr_1 >\mathcal{R} cr_i$.

Example 4. In our running example, we consider that gathering decision makers working at the same position cr_1 is the highest constraint because carried out studies demonstrate that the function is the most discriminating criteria in the multidimensional groupization in the data warehouse area.

B. Complex Ranking Constraints

The true power of constraint modeling comes with the advent of complex ranking constructors. Three complex constraints are studied: (i) equality of constraints presented as *pareto constraint*; (ii) *Prioritized constraint* expressing more important preference compared to others; (iii) *ranking-function of constraints*.

Definition 5. Pareto constraint: $cr_1 \otimes cr_2$

If cr_1 and cr_2 are equally important then $\mathcal{R} = > cr_1 \otimes cr_2$.

Example 5. In our running example, we consider that cr_1 is gathering the private managers and cr_2 is collecting the private investors. The constraints are equally important and are denoted $\mathcal{R} = > cr_1 \otimes cr_2$, because both of them are backboned on the function groupization criteria.

Definition 6. Prioritized constraint: $cr_1 \ominus cr_2$

If cr_1 is more important than cr_2 then $\mathcal{R} = > cr_1 \ominus cr_2$.

Example 6. In our running example, gathering the private managers cr_1 is a prioritized constraint compared to collecting responsibilities-based analysts cr_2 . Thus, we express such fact as follows $\mathcal{R} = > cr_1 \ominus cr_2$.

Definition 7. Ranking Preference function \mathcal{R}

$\mathcal{R} = (>rank_F (cr_1, cr_2))$

Example 7. In our running example, gathering the private managers denoted by cr_1 is more important than collecting source identification-based analysts cr_2 . Hence, we use a ranking constraint function, denoted as follows $\mathcal{R} = (>rank_F (cr_1, cr_2))$ to rank those constraints.

5 SHACUN Semi-supervised Hierarchical Active Clustering Based on ranking constraints algorithm

In this section, we describe our SHACUN Semi-supervised Hierarchical Active Clustering based on ranking constraints algorithm which is a new semi-supervised clustering method based on a semi-supervised hierarchical clustering process. Our method uses ranked constraints provided by expert along the iterations of agglomerative hierarchical semi-supervised clustering algorithm. In order to measure the similarity between objects, several similarity metrics exist. We devote the next subsection to this issue.

5.1 Similarity Metric

Several similarity measures are introduced in the hierarchical clustering to discover the closest pair of documents to merge. In our work, we learn a parameterized Jaccard similarity, because the Jaccard similarity is widely used for clustering text documents, on which we focused in our experiments. The Jaccard distance is defined as the size of the intersection of the two clusters C_i and C_j divided by the size of the union of those clusters.

$$Sim(C_i, C_j) = \frac{|C_i \cap C_j|}{|C_i \cup C_j|} \quad (1)$$

In our running example, we cluster OLAP log files of analysts where all generated queries of the each analyst are stored. Hence, we extend this measure to the multidimensional context. Hence, the inter-OLAP log files Jaccard distance is backboned on the MDX query structure, basically on similarity between used facts, measures, dimension attributes, as well as slicer specification members. The later is used in the WHERE clause and restricts the result data. Any dimension that does not appear on an axis in the SELECT clause can be named on the slicer. The similarity metric between two OLAP log files is given by the number of common queries having common facts, measures, dimensions and members of the common specification in both historical files, divided by the total number of queries in both files, it is computed using the following formula:

$$Sim(C_i, C_j) = \frac{C_{Queries(C_i, C_j)}}{\sum Queries(C_i) + \sum Queries(C_j) - C_{Queries(C_i, C_j)}}. \quad (2)$$

With $C_{Queries(C_i, C_j)}$: Number of common queries in two clusters (*i.e.* log files) C_i of analyst i and C_j of analyst j ,
 $\sum Queries(C_i)$: Sum of all existing queries in the cluster (*i.e.* log file) C_i . Let us consider two log files of two investors assigned to two clusters C_1 and C_2 , the first log file includes 4847 MDX queries and the second one contains 4982 MDX queries. The common queries in the two files is roughly 2269, the similarity between the two log files is computed using the jaccard distance adapted to the multidimensional context.

$$\begin{aligned} Sim(C_1, C_2) &= \frac{C_{Queries(C_1, C_2)}}{\sum Queries(C_1) + \sum Queries(C_2) - C_{Queries(C_1, C_2)}} \quad (3) \\ &= \frac{2269}{4847 + 4982 - 2269} = 0.3. \end{aligned}$$

5.2 SHACUN Process

In the following, we thoroughly discuss the phases of the SHACUN algorithm according to the pseudo-code shown by Algorithm 1. Starting from the data set, we propose the following phases to integrate ranked constraints in hierarchical clustering algorithm:

1. After the initialization, the treatment related to computing the similarity between objects is performed. The similarity matrix is output;
2. The expert ranks the constraints from highest to lowest important constraints, because he is the ablest to guide such clustering process;
3. The assignment consists in merging the most similar clusters in respect to the highest important constraint;
4. The update of the similarity matrix after merging clusters is consequently done.

Table 1. List of used notations in the *SHACUN* algorithm

Notation	Description
O	Set of objects
$>\mathcal{R}$	Set of ranked constraints
\mathcal{C}	Set of generated Clusters
η	Number of clusters
n_c	Current number of clusters
cr_m	The most important constraint
$Sim(C_i, C_j)$	Similarity distance between the two clusters C_i and C_j
<i>SimMatrix</i>	Similarity matrix

Algorithm 1. *SHACUN*: Semi-supervised Hierarchical Active Clustering based on ranking constraints

Data: $O, >\mathcal{R}, \eta$

Result: \mathcal{C} : Set of generated Clusters

```

1 begin
2   Set each object as cluster;
3    $n_c = count(O)$ ;
4   while ( $n_c <> \eta$ ) do
5     foreach  $C_i \in \mathcal{C}$  do
6       foreach  $C_j \in \mathcal{C}$  do
7         Compute  $Sim(C_i, C_j)$ ;
8         Store  $Sim(C_i, C_j)$  in SimMatrix;
9       // Function Highest outputs the most importance constraint
10       $cr_m = Highest(>\mathcal{R})$ ;
11      ( $C_i, C_j$ ) =  $cr_m$ ;
12      Merge( $C_i, C_j$ );
13       $n_c \leftarrow n_c - 1$ ;
13 end
14 Return  $\mathcal{C}$ ;

```

The used notations are depicted by table 1 and the pseudo-code of *SHACUN* is illustrated by the algorithm 1. In fact, *SHACUN*, an iterative process, operates in four successive steps. First, we consider each object as cluster, in our case study, each object is related to OLAP log file of decision-maker (cf. line 2). The current number of clusters is set to the number of objects. Then, we compute the similarity matrix *SimMatrix* using the Jaccard distance between all generated clusters (cf. lines 3-4). After that, we rank constraints from lowest to highest constraint into $>\mathcal{R}$. The pair of clusters that should be merged are derived using the *Highest* function. Indeed, this function takes as input the current ranking of constraints \mathcal{R} and outputs the two clusters that must be merged. Next, we merge the two clusters related to the most important constraints (cf. line 10) and we decrease the number of current clusters n_c . Finally, we update the similarity

matrix. This process is repeated until the number of generated clusters reaches the number of clusters η .

6 Experimental Results

To evaluate the effectiveness and efficiency of our algorithm *SHACUN*, we carried out extensive experiments. Indeed, we compare our approach with the pioneering algorithm falling within the hierarchical semi-supervised clustering trend. In order to assess the overall performance of *SHACUN* method, we carry out experiments on machine equipped with a 3 GHz Pentium IV and 2 GB of main memory. We report the experiments carried out on stock market data warehouse. Indeed, all analysts log files of the data warehouse are collected. Each log file contains approximately 5000 stored queries. An analytical study of the various factors that effectively identify groups of decision makers in the data warehouse leads to four criteria of groupization, namely: (i) the *function exerted*: we assume that analysts working in the same position have similar preferences, (ii) the *granted responsibilities* to accomplish defined goals: in fact two portfolio managers can not assume the same responsibilities, (iii) the *source of group identification*: it is explicit when an analyst specifies to which group he belongs otherwise this task may be implicitly performed, (iv) the *dynamic identification* of groups: the detected groups will be updated or will remain static.

Through the carried out experiments, we have a twofold aim: first, we have to stress on the assessment of *SHACUN* performances for each groupization criteria. Second, we focus on evaluating the overall performance of *SHACUN* approach *vs.* related approach in the literature. We choose to compare our algorithm to ID3 (Induction Decision Tree) [15] and NAIVE BAYES [7] classification methods using Weka platform 3.6.5 edition [9]. In our context of the groups identification, three key metrics may be used to assess the performance of our approach: (i) The True Positive rate (*TP*) measures the proportion of examples classified as class *X*, among all examples which truly belong to class *X*. It is equivalent to the recall; (ii) The False Positive rate (*FP*) measures the proportion of examples classified as class *X* then they belong to another class, whereas *TF* is related to the rate of accuracy; (iii) The receiver operating characteristic (*ROC*) [14] is the relationship between the rate of *TP* and *FP*.

6.1 *SHACUN* Performance Assessment

According to the predefined groupization criterion, we formulate our ranked constraints. For example, in the function context, it is more important to merge two clusters relative to two managers than to merge a cluster of managers and another one of investors. Such expressed constraints are conducted by the expert and dynamically change depending on the groupization criterion.

¹ <http://www.cs.waikato.ac.nz/ml/weka/>

Analysis Of The Function Criterion

First, we focus on the function criterion. The learned classes are: (i) portfolio managers, (ii) investors, (iii) private managers and (iv) private investors. Table 2 reports the different obtained results after varying the training sets according to the specified function. According to the ID3 technique, *SHACUN* engenders a *TP* equal to 68% for portfolio managers, 100% for investors, 100% for private managers and 100% for private investors.

However, with respect to NAIVE BAYES algorithm, *SHACUN* generates a *TP* equal to 62% for portfolio managers, 100% for investors, 100% for private managers and 100% for private investors. Note that the true positive rate of portfolio investors is the greatest because such function exerted focuses on a specific mission. Hence, the launched queries are usually repeated and are radically dissimilar from the other OLAP log files. However, each private manager or investor analyzes in a different way the socket market data warehouse. So, the generated log files include dissimilar queries that's why their contents are heterogeneous. Thus, the false positive rate is relatively high.

Table 2. True Positive *vs.* False Positive rate of *SHACUN* with respect to ID3 and NAIVE BAYES algorithm for function-based groupization

Training set	ID3		Naive Bayes	
	<i>TP</i>	<i>TF</i>	<i>TP</i>	<i>TF</i>
Logs of portfolio managers	0.68	0	0.62	0
Logs of investors	1	0.1	1	0.01
Logs of private managers	1	0	1	0
Logs of private investors	1	0	1	0.1

Analysis of the Granted Responsibility Criterion

We distinguish three classes according to granted responsibilities : (i) Short-term operational responsibilities, (ii) medium-term tactic responsibilities, (iii) long-term strategic responsibilities. Table 3 reports the accuracy of responsibilities-based groupization. In the context of operational responsibilities, the *SHACUN* engenders according to ID3 classification and NAIVE BAYES respectively true positive rate equal to 90% and 91%. For tactical responsibilities, the *SHACUN* induces a true positive rate equal to 64% with ID3 classification and generates a true positive rate equal to 61% with NAIVE BAYES classification. Regarding strategic responsibilities, the *SHACUN* provides respectively for ID3 and NAIVE BAYES algorithms true positive rate equal to 62% and 60%. From the results given in this table, we can conclude that, generally, the greatest true positive rate is performed by analysts having similar operational responsibilities. Indeed, decision-makers share similar preferences with respect to operational responsibilities. In fact, working on current projects, they will perform precise and focused analysis on data warehouse. Thus, the generated log files will be more similar in the context of accomplishing operational goals than strategic goals. This fact may explain the decrease of *TP* rate depending on the passage of time.

Table 3. True Positive *vs.* False Positive rate of *SHACUN* with respect to ID3 and NAIVE BAYES algorithm for responsibilities-based groupization

Training set	ID3		Naive Bayes	
	TP	TF	TP	TF
Logs collected with respect to operational responsibilities	0.9	0.21	0.91	0.2
Logs collected with respect to tactical responsibilities	0.64	0.01	0.61	0.02
Logs collected with respect to strategic responsibilities	0.62	0.01	0.60	0.02

Analysis of the Source Of Group Identification Criterion

The source of group identification can be either (i) implicit without any intervention of analyst, or (ii) explicitly depending on the choice of the analyst to which group he chooses to belong. As shown by table 4, for logs explicitly collected, according to both of ID3 and NAIVE BAYES algorithms, *SHACUN* generates true positive measure equal respectively to 85% and 82%. While, *SHACUN* for logs implicitly gathered provides a true positive metric equal to 94% for ID3 and 93.4% for NAIVE BAYES. We can see that the accuracy of *SHACUN* algorithm decreases when we implicitly learn the identified groups. This can be explained by the fact that it's better to choose the analyst's group explicitly than automatically learning it, because the analyst knows his colleagues and their preferences so he is the most skilled to select to which group belonging.

Table 4. True Positive *vs.* False Positive rate of *SHACUN* with respect to ID3 and NAIVE BAYES algorithm for source-based groupization

Training set	ID3		Naive Bayes	
	TP	TF	TP	TF
Logs implicitly collected	0.85	0.02	0.82	0.09
Logs explicitly collected	0.94	0.15	0.934	0.1

Analysis of the Dynamicity Criterion

The discovery of the groups can be either (i) *static* in a fixed way, or (ii) *dynamic* based on events that will raise. As reported by table 5, the *SHACUN* generates respectively for ID3 and NAIVE BAYES true positive rate equal to 84% and 82% for logs dynamically collected. While both algorithms produce true positive rate equal to 100% for statically gathered logs. This can be explained by the fact that dynamicity in groupization concerns circumstance variation and condition changing. So that, the evolved queries will radically change and the similarity of the log files will progressively reduce.

6.2 Overall Performance Of *SHACUN* Approach *vs.* Related Approaches In The Literature

To assess the performance of *SHACUN* algorithm, we choose to compare *SHACUN vs.* Klein et al.'s algorithm used in the groupization context. The

Table 5. True Positive *vs.* False Positive rate of *SHACUN* with respect to ID3 and NAIVE BAYES algorithm for dynamicity-based groupization

Training set	ID3		Naive Bayes	
	TP	TF	TP	TF
Logs statically collected	1	0.17	1	0.19
Logs dynamically collected	0.84	0	0.82	0

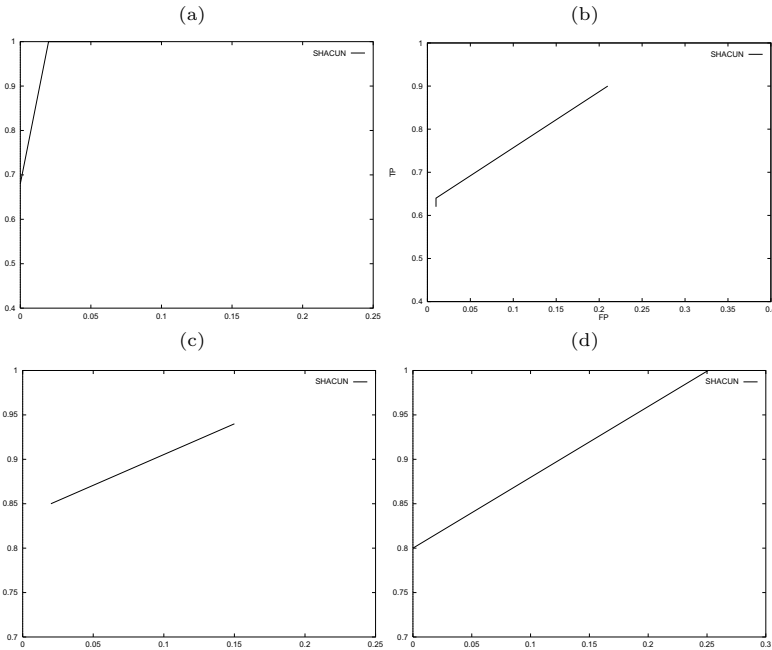


Fig. 3. Accuracy of *SHACUN* *vs.* Klein *et al.*'s algorithm with respect to ID3 classification according to the (a) **function-based groupization**, (b) **responsibilities-based groupization**, (c) **Source-based of groupization**, (d) **dynamicity-based groupization**

rates TP and FP change in relation to each other. Whenever the TP is the highest, the FP is the lowest, and vice versa. Consequently, these two metrics can be of use to plot a ROC curve (*Receiver Operating Characteristic*). Figure 3 compares the ROC curve of *SHACUN* *vs.* this of Klein's *et al.* one with respect to ID3 classification for the four groupization criteria. While figure 4 compares the ROC curve of *SHACUN* *vs.* this of Klein *et al.*'s algorithm with respect to NAIVE BAYES classification for the four groupization criteria.

The ROC curve assesses the accuracy of the *SHACUN*. Thus, we conclude that *SHACUN* is more accurate than Klein *et al.*'s algorithm. Our experimental results clearly showed the following interesting insights: (i) Adding ranked constraints can significantly reduce the number of distance calculations accordingly

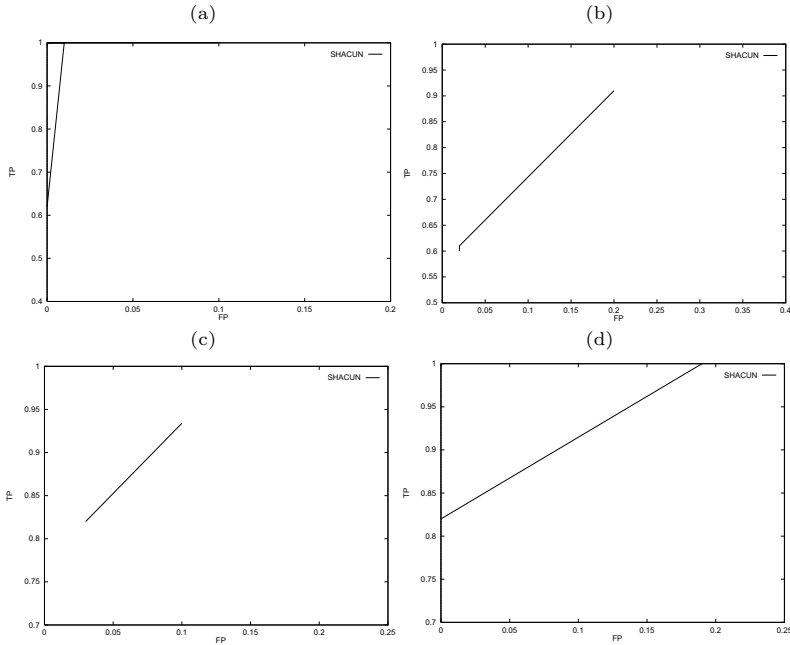


Fig. 4. Accuracy of *SHACUN* vs. Klein et al.'s algorithm with respect of NAIVE BAYES classifier according to the (a) **function-based groupization**, (b) **responsibilities-based groupization**, (c) **Source-based of groupization**, (d) **dynamicity-based groupization**

a significant improvement in accuracy is achieved; (ii) *SHACUN* algorithm is very beneficial for resolving the challenge on analysts groupization through enforcing ranked constraints.

7 Conclusion

In this paper, we proposed a new approach called *SHACUN*, permitting clustering derivation under ranked constraints. The main particularity of our approach is the application of ordinal constraints instead of boolean constraints. The carried out experimental results showed the effectiveness of the introduced algorithm and highlighted that *SHACUN* outperforms the pioneering algorithm in semi-supervised learning.

The preliminary obtained results offer exciting additional alternative avenues of future work. First, it will be interesting to make our method able to handle the uncertainty on the olap log files. Second, we are interested in extending our approach to consider missing data in collected log files.

References

1. Bade, K., Hermkes, M., Nürnberger, A.: User Oriented Hierarchical Information Organization and Retrieval. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 518–526. Springer, Heidelberg (2007)
2. Ben Ahmed, E., Nabli, A., Gargouri, F.: A Survey of User-Centric Data Warehouses: From Personalization to Recommendation. *The International Journal of Database Management Systems (IJDMS)* 3(2), 59–71 (2011)
3. Ben Ahmed, E., Nabli, A., Gargouri, F.: Building MultiView Analyst Profile From Multidimensional Query Logs: From Consensual to Conflicting Preferences. *The International Journal of Computer Science Issues (IJCSI)* 9(1), 124–131 (2012)
4. Bohm, C., Plant, C.: Hissclu: A hierarchical density-based method for semi-supervised clustering. In: Proceedings of the International Conference on Extending Database Technology (EDBT 2008), New York, USA, pp. 440–451 (2008)
5. Dasgupta, S., Ng, V.: Which clustering do you want? inducing your ideal clustering with minimal feedback. *Journal of Artificial Intelligence Research* 39, 581–632 (2010)
6. Davidson, I., Ravi, S.S.: Using instance-level constraints in agglomerative hierarchical clustering: theoretical and empirical results. *Data Mining and Knowledge Discovery* 18(2), 257–282 (2009)
7. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. *Machine Learning* 29, 131–163 (1997)
8. Golfarelli, M.: From User Requirements to Conceptual Design in Data Warehouse Design - a Survey. In: *Data Warehousing Design and Advanced Engineering Applications: Methods for Complex Construction*, pp. 1–16 (2008)
9. Huang, R., Lam, W.: An active learning framework for semi-supervised document clustering with language modeling. *Data and Knowledge Engineering* 68(1), 49–67 (2009)
10. Jain, A.K., Dubes, R.C.: Algorithms for clustering data. Prentice-Hall, Inc., Upper Saddle River (1988)
11. Kestler, H.A., Kraus, J.M., Palm, G., Schwenker, F.: On the Effects of Constraints in Semi-Supervised Hierarchical Clustering. In: Schwenker, F., Marinai, S. (eds.) ANNPR 2006. LNCS (LNAI), vol. 4087, pp. 57–66. Springer, Heidelberg (2006)
12. Klein, D., Kamvar, S.D., Manning, C.D.: From instance-level constraints to space-level constraints: making the most of prior knowledge in data clustering. In: *International Conference on Machine Learning (ICML 2002)*, pp. 307–314. Springer, San Francisco (2002)
13. Nogueira, B.M., Jorge, A.M., Rezende, S.O.: Hierarchical confidence-based active clustering. In: *The Symposium on Applied Computing*, pp. 535–537 (2012)
14. Provost, F., Fawcett, T.: The case against accuracy estimation for comparing induction algorithms. In: *International Conference on Machine Learning, Madison, Wisconsin USA*, pp. 445–453 (1998)
15. Quinlan, J.R.: Induction of decision trees. *Machine Learning*, 81–106 (1986)

A Minimum Spanning Tree-Inspired Clustering-Based Outlier Detection Technique

Xiaochun Wang¹, Xia Li Wang², and D. Mitch Wilkes³

¹ School of Electronics and Information, Xi'an Jiaotong University, Xi'an, 710049 China
xiaocchunwang@mail.xjtu.edu.cn

² Department of Computer Science, Changan University, Xi'an, 710061 China
xlwang@chd.edu.cn

³ School of Engineering, Vanderbilt University, Nashville, TN 37235 USA
mitch.wilkes@vanderbilt.edu

Abstract. Due to its important applications in data mining, many techniques have been developed for outlier detection. In this paper, an efficient three-phase outlier detection technique. First, we modify the famous k -means algorithm for an efficient construction of a spanning tree which is very close to a minimum spanning tree of the data set. Second, the longest edges in the obtained spanning tree are removed to form clusters. Based on the intuition that the data points in small clusters may be most likely all outliers, they are selected and regarded as outlier candidates. Finally, density-based outlying factors, LOF, are calculated for potential outlier candidates and accessed to pinpoint the local outliers. Extensive experiments on real and synthetic data sets show that the proposed approach can efficiently identify global as well as local outliers for large-scale datasets with respect to the state-of-the-art methods.

Keywords: distance-based outlier detection, density-based outlier detection, clustering-based outlier detection, minimum spanning tree-based clustering.

1 Introduction

Human species cannot survive without the ability to discover anomalous patterns in observed data that do not conform to the expected normal behavior. In statistics, the anomalous patterns are often referred to as outliers. In a classic definition, “an outlier is an observation that deviates so much from other observations that it aroused suspicions that it is generated by a different mechanism” [1]. Due to its important applications, outlier detection has found immense use in a wide variety of practical domains, such as intrusion detection for cyber-security [2,3], fraud detection for credit cards, insurance and tax [4], early detection of disease outbreaks in the medical field [5], fault detection in sensor networks for monitoring health, traffic, machine status, weather, pollution, surveillance [6], and so on [7,8].

As a result, the task of finding outliers in a data set has long been an area of active research, and different outlier detection techniques, such as distribution-based, depth-based, distance-based, density-based and clustering-based approaches, have been

developed and no techniques are completely satisfactory for all the situations. Originating from statistics, distribution-based methods study the outlier detection problem in the context of a given distributional model [9,10,11], which is usually not known a priori for modern large databases. As a result, they have limited usefulness. Stemming from Computational Geometry, depth-based methods organize the observations into layers through the computation of k -dimensional convex hulls, believing that the shallow layers are more likely to contain outliers than deep ones. But these methods are very computationally expensive for more than a few dimensions [12]. Distance-based methods and density-based methods actually address the detection problems of two different notions of outliers: more globally-oriented outliers (the former) and more locally-oriented outliers (the latter), and that clustering-based approaches obtain outliers as the by-products of clustering, that is, outliers are the data items that reside in the smallest clusters. Other types of outlier detection algorithms have also been developed to look into the problem from different aspects [13,14,15]. However, they are beyond the scope of this study and will not be discussed further here.

In this paper, we propose an in-memory fast outlier detection method which integrates an efficient minimum spanning tree (MST) based clustering algorithm [16] with outlier concepts [10] for modern large high-dimensional datasets. Basically, our CPU efficient MST-inspired outlier detection algorithm has three phases. In the first phase, a spanning tree very close to an MST is constructed. In the second phase, the longest edges in the spanning tree are identified and removed to form clusters as the standard MST-based clustering algorithms do, and the data points in those small clusters are selected as outlier candidates. In the third phase, the algorithm assigns an LOF to a small number of outlier candidates discovered in the second phase so as to utilize the density-based outlier detection technique to selectively mine the local outliers. Our contributions include:

- Extensive experimental evaluation on both synthetic and real datasets demonstrates the meaningfulness of our approach as a very efficient way for the detection of global and local outliers.
- The proposed method combines the advantages of distanced-based, density-based and clustering-based outlier detection techniques to give a better intuition to view such techniques.
- Compared to the state-of-art distance-based algorithms which need some parameters to be provided by the user, the proposed outlier detection methods overcomes this limitation, thus proving to be an effective solution in real applications where a completely unsupervised method is desirable.
- As far as local outliers are concerned, LOF is computationally expensive. However, our algorithm checks distance ranking and only calculates LOF when distances are distributed evenly, that is, when there is not very much density differences. By this way, only a relatively small computation is sufficient for preserving the good quality of the detection results.

The rest of the paper is organized as follows. In Section 2, we review some existing work on distance-based, density-based and clustering-based outlier detection

algorithms. We next present our proposed approach in Section 3. In Section 4, an empirical study is conducted to evaluate the performance of our algorithm with respect to some state-of-the-art outlier detection algorithms. Finally, conclusions are made and future work is indicated in Section 5.

2 Related Work

There are three parts of the unsupervised outlier detection literature that are related to our study: distance-based outlier detection, density-based outlier detection and clustering-based outlier detection.

2.1 Distance-Based Outlier Detection

Proposed by Knorr and Ng, distance-based outlier detection methods provide a good way to detect outliers residing in relatively sparse regions. Given a distance measure on a feature space, the notion of outliers studied by Knorr and Ng is defined as: “An object O in a dataset T is a distance-based outlier, denoted by $DB(p, D)$ -outlier, if at least a fraction p of the objects in T lies greater than distance D from O , where the term $DB(p, D)$ -outlier is a shorthand notation for a Distance-Based outlier (detected using parameters p and D)” [11]. Beginning with this work, various versions of distance-based outlier definition have been developed. Three popular ones are:

1. Given a real number d and an integer p , a data item is an outlier if there are fewer than p other data items within distance d [11,12].
2. Given two integers, n and k , outliers are the data items whose distance to their k -th nearest neighbor is among top n largest ones [17].
3. Given two integers, n and k , outliers are the data items whose average distance to their k nearest neighbors is among top n largest ones [18,19].

The first definition does not give a ranking but requires the specification of a distance parameter d , which may involve trial and error to guess an appropriate value [17]. Eliminating this requirement, the second definition only considers the distance to the k -th nearest neighbor and ignores information about closer points. The last definition accounts for the distances to k nearest neighbors and, thus, is slower to calculate than the first two.

In [11], three approaches were proposed for $DB(p, D)$ -outliers: a $O(dN^2)$ block-oriented nested loop algorithm, a $O(M\log N)$ index-based algorithm for low dimensions [10] and cell-based algorithm (with a time complexity increasing exponentially with the dimensionality). However, these algorithms can be used to detect outliers efficiently only for low dimensional data sets. Theoretically, if a (reasonably tight) cutoff threshold can be determined efficiently, for most normal data, a partial search through the database is enough to determine it is not an outlier. Only for data objects that are potential outlier candidates can a full scan through the database be necessary. Based on this idea, distance-based outlier detection algorithms proposed thereafter (ORCA method [20], RBRP method [21], DHCA-based methods [22,23]) have been

focusing on determining the cutoff threshold efficiently and reducing the partial search sufficiently, by randomizing data and/or using some data structures.

2.2 Density-Based Outlier Detection

Distance-based outlier detection techniques work well for detecting global outliers in simply-structured data sets that contain one or more clusters with similar density. However, for many real world data sets which have complex structures in the sense that different portions of a database can exhibit very different characteristics, they might not be able to find all interesting outliers. A classic two-dimensional illustration to show this deficiency is shown in Fig. 1.

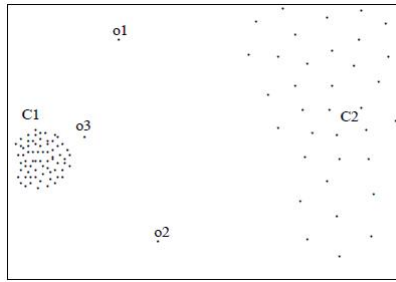


Fig. 1. Two clusters with different densities

This data set contains a dense cluster, C_1 , a sparse cluster, C_2 , and three outstanding objects. Obviously, the two well separated objects o_1 and o_2 are outliers no matter whether we look from a global or local point of view. However, according to any distance-based outlier definitions, object o_3 can not be detected as an outlier, without all the objects in cluster C_2 being detected as outliers, though, from a more local point of view, it should be detected as an outlier. To deal with this situation, Breunig et al. pioneered the density-based outlier detection research by assigning to each object a degree of being an outlier, called the Local Outlier Factor (LOF), for judging the outlyingness of every object in the data set based on ratios between the local density around an object and the local density around its neighboring objects [10].

The LOF method works by first calculating the LOF for each object in the data set. Next, all the objects are ranked according to their LOF values. Finally, objects with top- n largest LOF values are marked as outliers. Calculating an exact LOF for each data object can be computationally expensive. To solve this problem, Jin et al. proposed to use the concept of micro-clusters to efficiently mine top- n LOF-based outliers in large databases [24]. As a further extension, the algorithm presented in [25] uses the reverse nearest neighbors additionally and considers a symmetric relationship between both values as a measure of outlyingness. Several other extensions and refinements have been proposed, including a Connectivity-based Outlier Factor (COF) [26], Local Outlier Integral (LOCI) [9], and a Spatial Local Outlier Measure (SLOM) [27]. The main difference between LOF and LOCI is that the former uses k nearest neighbors while the latter uses ϵ -neighborhoods.

2.3 Clustering-Based Outlier Detection

A problem associated with distance-based as well as density based outlier detection algorithms is their strong sensitiveness to the setting of some parameters. This can be illustrated by a 2-dimensional data set shown in Fig. 2. For distance-based outlier detection techniques, if $k = 6$ nearest neighbors are considered, all the points in cluster C3 will not be detected as outliers, while if $k = 7$, all the data points in cluster C3 are regarded as outliers. Similar problems exist for density-based outlier detection techniques. The situation could be worse for the detection of outliers in high-dimensional feature space since data points cannot be visualized there.

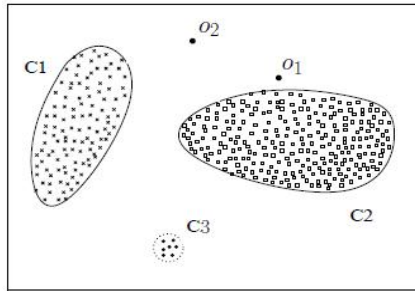


Fig. 2. Sample clusters in a 2-D data set

This is where clustering algorithms can be of some help. Being a very important data mining tool, the main concern of clustering algorithms is to find clusters by optimizing some criterion, such as minimizing the intra-cluster distance and maximizing the inter-cluster distance. As a by-product, data items in small groups can often be regarded as outliers (noise) that should be removed to make clustering more reliable. Classic clustering algorithms, such as k -means algorithm or PAM, rely on grouping the data points around some “centers” and do not work well when the boundaries of the clusters are irregular. As an alternate, graph theory based methods, typified by the MST-based clustering algorithms, can find clusters with irregular boundaries.

A minimum spanning tree is a connected weighted graph with no closed paths that contains every point in the data set and has the minimal total weight. If a weight denoting a distance between two end points is assigned to each edge, any edge in an MST will be the shortest distance between two subtrees that are connected by that edge. This fact is referred to as the cut property of MSTs. Therefore, removing the longest edges corresponds to choosing the breaks to form clusters. Based on this finding and initially proposed by Zahn [28], MST-based clustering algorithms have so far been extensively studied [17]. Regarding data points in smallest clusters formed by cutting the longest edges in an MST may likely be outliers, several MST-based outlier detection techniques have been proposed [29,30,31]. But these MST-based clustering algorithms are very computationally expensive (from $O(N \log N)$ to $O(N^2)$).

Recently, an MST-based clustering algorithm with a near linear performance has been developed [16], and can be modified to suit for our purpose. Basically, their

CPU efficient MST-based clustering algorithm has three phases. First, a simple spanning tree is constructed. Secondly, in order to quickly identify the longest edges, a modification of the classic K -means algorithm, called divisive hierarchical clustering algorithm (DHCA), is used to generate a spanning tree very close to a true minimum spanning tree. Finally, the cut and the cycle property of the minimum spanning trees are used to perform the clustering. The main advantage of the algorithm is its computation efficiency.

For a given data set, DHCA starts with K randomly selected centers, and assigns each point to its closest center, creating K partitions. Then, for each of these K partitions, it recursively selects K random centers and continues the clustering process within each partition to form at most K^n partitions for the n th stage. In our implementation, the procedure continues until the number of elements in a partition is below $K + 2$, at which time, the distance of each data item to other data items in that partition can be updated with a smaller value via a brute-force nearest neighbor search. Such a strategy ensures that points that are close to each other in space are likely to be collocated in the same partition. A detailed demonstration and proof of the effectiveness of DHCA on k -nearest neighbors search has been given in [16].

3 An Improved MST-Based Outlier Detection Algorithm

According to the cut property, MST-based clustering algorithms can be very useful in detecting small clusters that can be selected and regarded as outliers. For example, after removing the first three longest edges of an MST constructed for the data set shown in Fig. 1, four clusters result and global outliers such as o1 and o2 can be immediately recognized as small clusters. Even for local outlier o3, after clustering, it can be detected as a global outlier from the viewpoint of cluster C1. However, most MST-based clustering algorithms require a complete MST be constructed in the first step. For modern databases typically consisting of millions of high dimensional data items, this can be very computationally expensive. Motivated by the approach presented in [17], in this section, we describe a new scheme to facilitate efficient MST-based outlier detection for modern large databases, which is based on the observation that, for some MST-based clustering approaches, if we can find the longest edges in an MST very quickly, there is no need to compute the exact distance values associated with the shortest ones. Therefore, for cases where the number of the longest edges that separate the clusters (including the noisy ones) can be much fewer than the number of the shorter ones (e.g., several dense clusters and a small number of outlier groups), MST-based outlier detection can be more efficient if the longest edges can be identified quickly before most of the shorter ones are found, which allows us to reduce the number of distance computations to a value smaller than $O(N^2)$.

3.1 A Simple Idea

Based on the spanning tree (which is very close to a true MST) constructed following the first two stages of the approach presented in [16] (i.e., sequential initialization and DHCA updates), we then search for the edge that has the largest value, called the

potential longest edge candidate, and cut it to form two partitions. If the number of data items in one partition is no more than p , we check whether there exists another edge with a smaller weight crossing the two partitions connected now by this potential longest edge candidate. This can be done by no more than $p(N-p)$ number of distance computations. If the result shows that there exists no such edge, we declare these data items to be outliers. Otherwise, we record the update (i.e., replace the potential longest edge candidate with the edge of a smaller weight crossing the two partitions) and start another search for the longest edge candidate in the MST. The point is that the current potential longest edge candidate is the longest edge in the tree so far that connects the only partition with no more than p data items to one of the partitions with more than p data items and provides an upper bound to the distance between the two partitions now connected by it. Since the number of outliers is expected to be relatively small, the number of distance computations consumed is expected to be relatively small as well. The working idea behind our efficient MST-based outlier detection algorithm is that some of the longest edges do not correspond to any cluster separations or breaks but are associated with the outliers.

3.2 Detecting Local Outliers

As mentioned previously, data sets under consideration may have complex structures in the sense that different portions of a database can exhibit very different density characteristics. For these cases, the measure of outlyingness only on simple distances between data points is not sufficient. As a further improvement, we take a step further by calculating LOF for each outlier candidate until the desired number of outliers are discovered. The LOF computation for every data point can be expensive. Fortunately, many indexing structures proposed in recent years for higher dimensional feature space can be utilized [32]. We are particularly interested in one called iDistance [39], which consists of three steps. First, the data are clustered into a set of partitions. Second, a reference point is identified for each partition. Finally, all data points are represented into a single dimensional space by indexing them based on the distance from the nearest reference point. In our approach, we choose the origin as the reference point. When the database is read in, the distance of each data item to the reference point is calculated. Next, the data items are sorted according to their distances. To be used as a search structure, given a data point, the search starts from its position in the sorted distances and proceeds bi-directionally along the radius axis until the k^{th} nearest neighbor is found. Thus, the use of the search structure can reduce the $O(N)$ time full search to a faster partial search. It is easy to see that the search structure can be constructed by any one dimensional sorting algorithm.

3.3 Our MST-Clustering Based Outlier Detection Algorithm

Our MST-based outlier detection algorithm can be summarized in the following:

1. initialize a spanning tree sequentially, that is, each data item can be assigned the distance between itself and its immediate predecessor.

4. refine the spanning tree by running DHCA multiple times until the percentage difference between two consecutive tree weights is below a threshold, say 0.001.
5. identify the potential longest edge candidate and remove it to form new clusters in a recursive fashion until the number of data items in the smallest cluster is below a predefined size of the largest outlier cluster (denoted by p here).
6. for each data item in the smallest cluster, calculate its distance to every other data item that is not in its cluster, which is $N-p$ number of distance computations.
7. if no edges with a smaller weight crossing the smallest cluster to the rest of data set exist, we declare these data items to be outlier candidates, otherwise we update each data point to its nearest neighbor in the rest of data set.
8. calculate a LOF for each outlier candidates and rank the LOF scores.
9. if a required number of outliers are found, stop, otherwise, start another round of search, i.e., go to 3

To summarize, the numerical parameters the algorithm needs from the user include the data set, the loosely estimated numbers of outliers, the input K to DHCA, and the number of nearest neighbors, k , for LOF calculation, while the outputs will be the ranked outliers. From the algorithm description, we expect the time spent on the first phase (including the sequential initialization and the DHCA updates) to scale as $O(fN\log N)$, where f denotes the number of DHCA's constructed. The second phase partitions the obtained approximate minimum spanning tree to locate the potential outliers. Since the number of desired outliers is much smaller than the data set size N , we expect the second phase to scale as $O(Ne)$, where e denotes the number of data points checked. We expect the third phase to scale as $O(ek^2\log N)$, where k denotes the number of nearest neighbors predefined to calculate LOF. Therefore, the average time complexity of our algorithm is $O(fN\log N + Ne + ek^2\log N)$, though the worst case could still be $O(N^2)$.

4 A Performance Study

In this section, we present the results of an experimental study performed to evaluate our MST-clustering based outlier detection algorithm. First, we select four 2-dimensional outlier detection problems and compare the performance of our proposed algorithm to that of the MST-based clustering method, the ORCA method and the LOF method. For this comparison, we would like to show that our MST-inspired clustering-based algorithm can outperform the classic outlier detection algorithms in both the execution time and the classification accuracy. We also study the behavior of the proposed algorithm with various parameters and under different workloads. Finally, we evaluate our algorithm on several real higher dimensional large data sets with no assumptions made on the data distribution and compare it with three state-of-the-art outlier detection algorithms to check the technical soundness of this study. All the data sets are briefly summarized in Table 1.

We implemented all the algorithms in C++ and performed all the experiments on a computer with Intel Core 2 Duo Processor E6550 2.33GHz CPU and 2GB RAM. The operating system running on this computer is Ubuntu Linux. We use the timer utilities

defined in the C standard library to report the CPU time. In our evaluation, we use the total execution time in seconds and the accuracy of the detected outliers as the performance metric. Each result we show was obtained as the average value over 10 runs of the program for each data set. In all the experiments, the total execution time account for all three phases of our MST-inspired outlier detection algorithm. The results show the superiority of our MST-inspired algorithm over the other algorithms.

Table 1. Descriptions of all data sets

Data Name	Data Size	Dimension	# of outliers	RT(s) of MST	RT(s) of LOF
Data11	10,180	2	28	23	N/A
Data12	20,360	2	56	88	N/A
Data13	30,540	2	84	198	N/A
Data21	11,550	2	30	28	N/A
Data22	23,100	2	60	114	N/A
Data23	34,650	2	90	253	N/A
Data31	16,165	2	34	N/A	89
Data32	32,330	2	68	N/A	360
Data33	48,495	2	102	N/A	798
Data41	20,066	2	36	N/A	137
Data42	40,132	2	72	N/A	546
Data43	60,198	2	108	N/A	1215
Corel	68,040	32	N/A	4,771	13,947
IPUMS	88,443	61	N/A	14,992	43,032
ourData	65,798	10041	N/A	31,610	51,938
Coverttype	581,012	55	N/A	602,911	N/A

4.1 Performance of Our Algorithm on Synthetic Data

In this subsection, we investigate the relative performance of our proposed algorithm on four outlier detection tasks shown in Fig. 3. Data11 contains two clusters and 28 global outliers. Data21 contains two curving irregularly shaped bands, and 30 global outliers. To clearly demonstrate the advantage of our algorithm over classic distance-based outlier detection algorithms on large data sets, we make the following assumptions for these two tasks: the densities of the clusters are similar. Data31 contains two

clusters of different densities, with 219 data points in the lower-density cluster, and some local outliers. Data41 contains two clusters of different densities as well, with 1581 data points in the lower-density cluster, and some local outliers. To test the running time (RT) scalability of our algorithm with the size of the data sets, we double and then triple these data sets to form their 4-cluster versions (Data12, Data22, Data32 and Data42) and 6-cluster versions (Data13, Data23, Data43 and Data33), respectively.

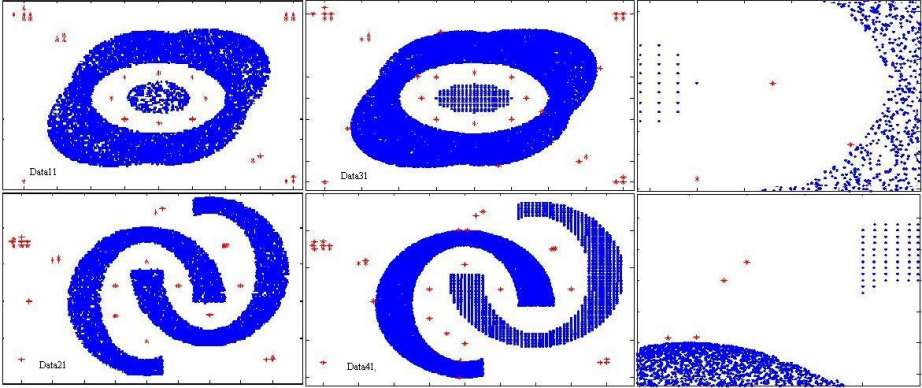


Fig. 3. (a) Data11 and Data21, (b) Data31 and Data41, (c) two snapshots of local outliers

We first study the effect of K , the input to the DHCA, and the impact of the dimensionality on the performance of our algorithm. Setting the largest number of data points in an outlying group to be 10, we varied K from 3 to 30 for Data13 and Data23.

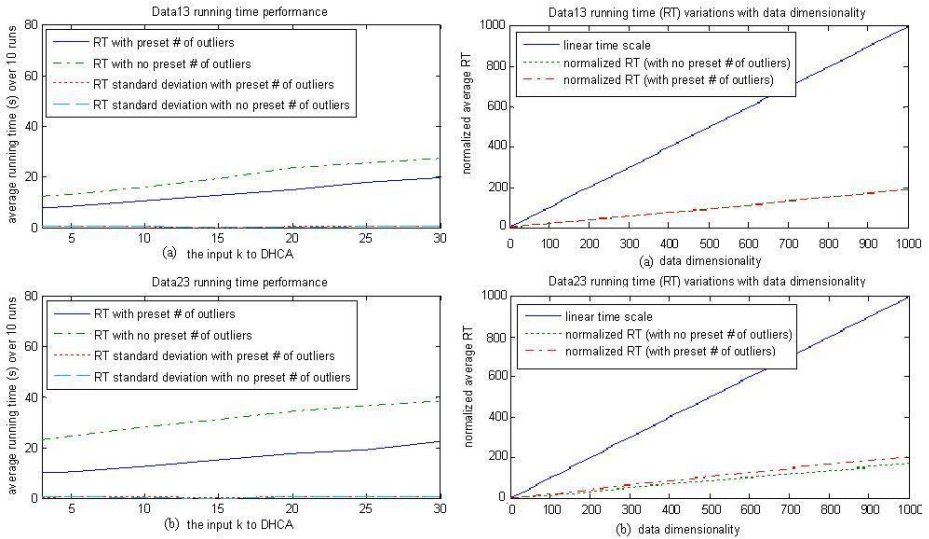


Fig. 4. Impacts of (left) K to DHCA (right) dimensionality on our algorithm

From the graphs on the left side of Fig. 4, we can see the impact of input to DHCA, K , on the efficiency of our algorithm. Overall, when K is small, the overhead of constructing the DHCA dominates. For large K , more distance computations to the partition centers are involved and the increases in the distance computations eventually dominate. Since the number of clusters and the longest edges are relatively small compared to the number of data points in the data sets, most nearest neighbor distances obtained using DHCA are much smaller than the longest ones and, therefore, our algorithm has a better scalability to the data size than the other algorithms. To see the impact of the dimensionality of data on our algorithm, we increase the data dimensionality by appending each data point to itself to result in a similarly distributed data set of a higher dimension. The results in terms of the running time (RT) on the first two data sets with extended dimensionality are shown on the left of Fig. 4. The runs were done to mine the outliers for $K = 5$ and the largest number of data points in an outlying group being set to be 10. It can be seen that all the running time increases with the size of the dimensionality in a linear fashion.

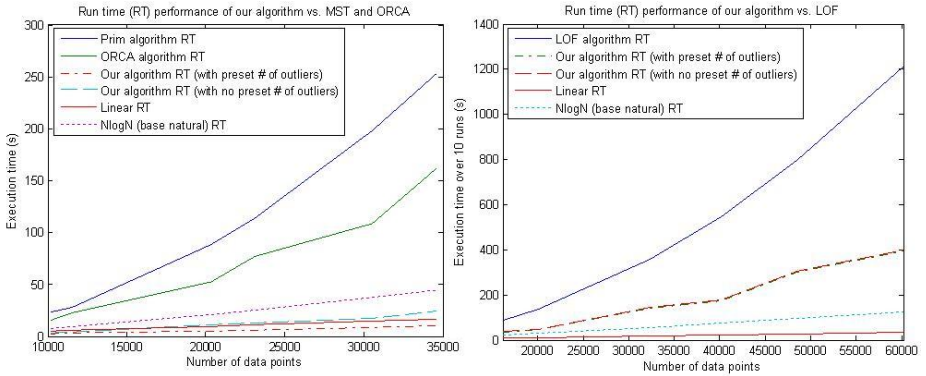


Fig. 5. Run time performance of our algorithm vs. (left) MST & ORCA (right) LOF

We next investigate the relative performance of our algorithm with respect to the classic MST- clustering based algorithm and the ORCA algorithm with/without the number of outliers given on Data11 through Data23. This set of experiments is run with K for the DHCA being set to be 5 and the largest number of data points in an outlying group being set to be 10. If the number of outliers is not given beforehand, our algorithm stops when the performance levels off, i.e., when there appears a big jump (between the $(n-1)^{th}$ and the n^{th} longest edges in this experiment) as measured by above 50% of the $(n-1)^{th}$ longest edge. This is a direct result of our assumptions made upon the data distribution. The running time results are presented on the left of Fig. 5. We then investigate the relative performance of our proposed algorithm with respect to the LOF algorithm on Data31 through Data43 when some local outliers also exist. The running time performance is shown on the right of Fig. 5.

In both graphs, the top lines represent the running time of the Prim’s MST algorithm and that of the LOF algorithm, respectively, and, clearly, they increase with the

data set sizes in a quadratic form. The expected execution time to find the small number of outliers given an $M\log N$ time algorithm and a linear time algorithm are extrapolated from the running time consumed by the Prim's algorithm and the LOF method, respectively. From the left figure, it can be seen that our algorithm outperforms the Prim's algorithm by an order of magnitude in running time and exhibits near linear scalability with the data sizes with 100% correct detection rate. From the right figure, it can be seen that our algorithm outperforms the LOF algorithm by a factor between 3.0 and 4.0 with 100% correct detection rate.

4.2 Performance of Our Algorithm on Real Data

In this subsection, we investigate the relative performance of our proposed algorithm on four real data sets. This set of experiments is run to mine top 100 outliers. First, we show the impact of the input K to DHCA on the run time performance. Setting the largest number of data points in an outlying group to be 30, we varied K from 3 to 30. The results are shown on the left side of Fig. 6 and agree with our observation for the synthesis data sets.

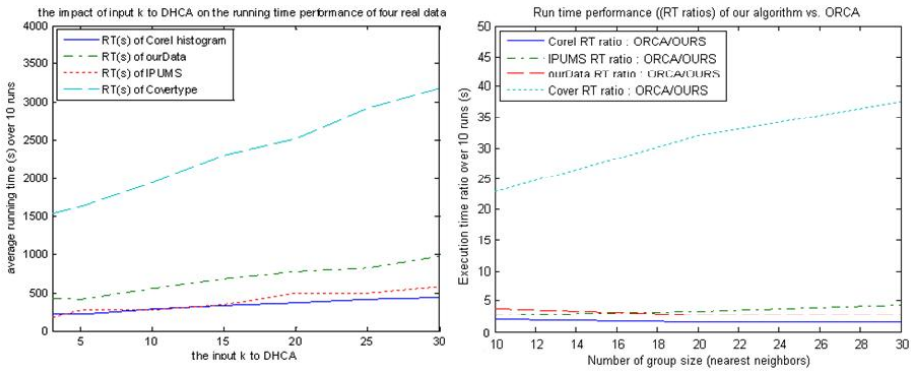


Fig. 6. (left) Impacts of K to DHCA (right) run time performance of our algorithm vs. ORCA

Setting the input value of K for the DHCA to be 5, we varied the largest number of data points in an outlying group from 10 to 30 to test the run time performance of our algorithm against the ORCA method. From the results shown on the right of Fig. 6, we can see our algorithm outperforms the ORCA method in all group sizes (or k , the number of nearest neighbors). Also we observe that the running time of our algorithm on ourData actually decreases with the increase of the outlying group size. This is intuitive because, when the outlying group size is set small, groups with a little larger size will be regarded as non-outlying groups, showing the advantage of clustering based outlier detection technique.

The run time performance of our algorithm vs. the MST-clustering based algorithm for the largest group size (i.e., 30) is summarized in Table 2, together with the $O(M\log N)$ (base natural) run time, which is extrapolated from the running time

consumed by the MST-clustering based method. From the table, we can see that our algorithm outperforms MST by an order of magnitude (with 100% correct detection rate) but is near the $O(M\log N)$ running time performance.

Then we investigate the relative performance of our proposed algorithm with respect to the LOF method on the first three real data sets. This set of experiments is run to mine top 100 outliers with the largest number of data points in an outlying group being set to 30. The results are shown in Table 3. From the table it can be seen that our algorithm outperforms the LOF method and has better performance when the dimensionality of the data set increases. In other words, the distance computation eventually dominates our algorithm for higher and higher dimensional data sets.

Table 2. Run time performance of OURS vs. MST

Data Name	MST/OURS	$N\ln N$ /OURS
Corel histogram	9.6	0.59
IPUMS	54.9	2.15
OURS	44.1	1.29
Coverttype	193.4	1.66

Table 3. Run time performance of OURS vs. LOF

Data Name	LOF/OURS	Dimensionality	Data Size
Corel histogram	1.6	32	68,040
IPUMS	3.2	61	88,443
ourData	5.1	90	65,798

Finally, the agreement of number of outliers detected using the MST-clustering based method, the ORCA method, the LOF method and our method are summarized in Table 4.

Table 4. Agreement of the number of outliers between different methods

Data Name	MST/OURS	ORCA/OURS	LOF/OURS
Corel histogram	100%	53%	73%
IPUMS	100%	59%	70%
OURS	100%	71%	65%

5 Conclusion

As a graph partition technique, MST-based clustering algorithms are of growing importance in detecting outlier clusters. A central problem in such applications in large high-dimensional data mining is usually its $O(N^2)$ time complexity. In this paper, we have presented a new MST-inspired outlier detection algorithm for large data sets by utilizing a divisive hierarchical clustering algorithm which makes the implementation

of the clustering-based outlier detection more efficiently. Additionally, our algorithm can be combined with density-based outlier concept to mine local outliers.

We conducted an extensive experimental study to evaluate our algorithm against the state-of-the-art outlier detection algorithms. Our experimental results show that our proposed MST-inspired clustering-based outlier detection algorithm is very effective and works reasonable well on all the data sets presented. However, the issue still remains that our algorithm, the ORCA method and the LOF method are based on different outlier definitions. From our observation, there exist some agreements as well as differences on the retrieved top outliers between these definitions. Since there often exist some structures in the data sets, the terminating condition of our algorithm should be further studied to unify all the outlier definitions.

References

1. Hawkins, D.M.: Identification of Outliers, Monographs on Applied Probability and Statistics. Chapman and Hall, London (1980)
2. Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S.: A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In: Data Mining for Security Applications (2002)
3. Lane, T., Brodley, C.E.: Temporal sequence learning and data reduction for anomaly detection. *ACM Transactions on Information and System Security* 2(3), 295–331 (1999)
4. Bolton, R.J., David, J.H.: Unsupervised Profiling Methods for Fraud Detection. *Statistical Science* 17(3), 235–255 (2002)
5. Wong, W., Moore, A., Cooper, G., Wagner, M.: Rule-based Anomaly Pattern Detection for Detecting Disease Outbreaks. In: Proceedings of the 18th National Conference on Artificial Intelligence (2002)
6. Sheng, B., Li, Q., Mao, W., Jin, W.: Outlier detection in sensor networks. In: Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 219–228 (2007)
7. Hodge, V.J., Austin, J.: A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review* 22, 85–126 (2004)
8. Chandola, V., Banerjee, A., Kumar, V.: Anomaly Detection: A Survey. *ACM Computing Surveys* 41(3), article 15 (2009)
9. Gibbons, P.B., Papadimitriou, S., Kitagawa, H., Christos Faloutsos, C.: LOCI: Fast Outlier Detection Using the Local Correlation Integral. In: Proceedings of the IEEE 19th International Conference on Data Engineering, Bangalore, India, pp. 315–328 (2003)
10. Breuning, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: Identifying Density-Based Local Outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 93–104 (2000)
11. Knorr, E.M., Ng, R.T.: Algorithms for Mining Distance-Based Outliers in Large Datasets. In: Proceedings of the 24th VLDB Conference, New York, USA, pp. 392–403 (1998)
12. Knorr, E.M., Ng, R.T.: Finding intensional knowledge of distance-based outliers. In: Proceedings of the 25th VLDB Conference, Edinburgh, Scotland, UK, pp. 211–222 (1999)
13. Angiulli, F., Pizzuti, C.: Outlier mining in large high dimensional datasets. *IEEE Transactions on Knowledge and Data and Engineering*, 203–215 (2005)
14. Niu, K., Huang, C., Zhang, S., Chen, J.: ODDC: outlier detection using distance distribution clustering. In: HPDMA 2007 in Conjunction with PAKDDd 2007, pp. 332–343 (2007)

15. Kreigel, H.P., Schubert, M., Zimek, A.: Angle-based outlier detection in high-dimensional data. In: *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, Nevada, USA, pp. 444–452 (2008)
16. Wang, X., Wang, X.L., Wilkes, D.M.: A Divide-And-Conquer Approach For Minimum Spanning Tree-Based Clustering. *IEEE Transactions on Knowledge and Data Engineering* 21(7), 945–958 (2009)
17. Knorr, E.M., Ng, R.T., Tucakov, V.: Distance-based outliers: algorithms and applications. *VLDB Journal: Very Large Databases* 8(3-4), 237–253 (2000)
18. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. In: *Proceedings of the ACM SIGMOD Conference*, pp. 427–438 (2000)
19. Angiulli, F., Pizzuti, C.: Fast outlier detection in high dimensional spaces. In: *Proceedings of the Sixth European Conference on the Principles of Data Mining and Knowledge Discovery*, pp. 15–26 (2002)
20. Bay, S.D., Schwabacher, M.: Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: *KDD 2003*, pp. 29–38 (2003)
21. Ghoting, A., Parthasarathy, S., Otey, M.E.: Fast mining of distance-based outliers in high-dimensional datasets. In: *SDM 2006*, pp. 608–612 (2006)
22. Wang, X., Wang, X.L., Wilkes, D.M.: A fast distance-based outlier detection technique. In: *Poster and Workshop Proceedings of 8th Industrial Conference on Data Mining*, Leipzig, Germany, pp. 25–44 (July 2008)
23. Wang, X., Wang, X.L., Wilkes, D.M.: Application of two partial search methods to Euclidean distance-based outlier detection. In: *Proceedings of the 2008 International Conference on Data Mining*, Las Vegas Nevada, USA, July 2008, pp. 420–426 (2008)
24. Jin, W., Tung, A.K.H., Han, J.: Mining top-n local outliers in large databases. In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, California, USA, pp. 293–298 (2001)
25. Jin, W., Tung, A.K.H., Han, J., Wang, W.: Ranking Outliers Using Symmetric Neighborhood Relationship. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) *PAKDD 2006*. LNCS (LNAI), vol. 3918, pp. 577–593. Springer, Heidelberg (2006)
26. Tang, J., Chen, Z., Fu, A.W.-c., Cheung, D.W.: Enhancing Effectiveness of Outlier Detections for Low Density Patterns. In: Chen, M.-S., Yu, P.S., Liu, B. (eds.) *PAKDD 2002*. LNCS (LNAI), vol. 2336, p. 535. Springer, Heidelberg (2002)
27. Sun, P., Chawla, S.: On local spatial outliers. In: *Proceedings of the 4th International Conference on Data Mining (ICDM)*, Brighton, UK (2004)
28. Zahn, C.T.: Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers* C-20, 68–86 (1971)
29. Rohlf, F.J.: Generalization of the gap test for the detection of multivariate outliers. *Biometrics* 31, 93–101 (1975)
30. Jiang, M.F., Tseng, S.S., Su, C.M.: Two-Phase Clustering Process for Outliers Detection. *Pattern Recognition Letters* 22, 691–700 (2001)
31. Lin, J., Ye, D., Chen, C., Gao, M.: Minimum Spanning Tree Based Spatial Outlier Mining and Its Applications. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) *RSKT 2008*. LNCS (LNAI), vol. 5009, pp. 508–515. Springer, Heidelberg (2008)
32. Yu, C., Ooi, B.C., Tan, K.L., Jagadish, H.V.: iDistance: An adaptive B+tree based indexing method for nearest neighbor search. *ACM Transactions on Data Base Systems (TODS)* 30(2), 364–397 (2005)

ML-DS: A Novel Deterministic Sampling Algorithm for Association Rules Mining*

Samir A. Mohamed Elsayed**, Sanguthevar Rajasekaran, and Reda A. Ammar

Computer Science Department, University of Connecticut

Abstract. Due to the explosive growth of data in every aspect of our life, data mining algorithms often suffer from scalability issues. One effective way to tackle this problem is to employ sampling techniques. This paper introduces, ML-DS, a novel deterministic sampling algorithm for mining association rules in large datasets. Unlike most algorithms in the literature that use randomness in sampling, our algorithm is fully deterministic. The process of sampling proceeds in stages. The size of the sample data in any stage is half that of the previous stage. In any given stage, the data is partitioned into disjoint groups of equal size. Some distance measure is used to determine the importance of each group in identifying accurate association rules. The groups are then sorted based on this measure. Only the best 50% of the groups move to the next stage. We perform as many stages of sampling as needed to produce a sample of a desired target size. The resultant sample is then employed to identify association rules. Empirical results show that our approach outperforms simple randomized sampling in accuracy and is competitive in comparison with the state-of-the-art sampling algorithms in terms of both time and accuracy.

1 Introduction

Data mining is the process of discovering patterns from large data sets. Mining algorithms often require multiple passes over the full dataset which can be a performance challenge due to the ever-increasing volume of data. One way to tackle this scalability issue is to use only a sample of the data. However, sampling produces only approximate results. There is usually a trade-off between sampling ratio and the desired accuracy. Naturally, the larger the sampling ratio, the higher will be the accuracy.

Agrawal, et al. [2] proposed association rules mining for market basket data patterns. Association rules identify correlations among a set of items found in transactions. The input to the association rules mining problem is a set of transactions where each transaction is a set of items. A set of items is also referred to as an *item-set*. A *k*-item-set is an item-set of size *k*. There are two measures proposed in [2] that quantify the significance of an association rule, namely, support

* This work is partially supported by the following grants: NSF0829916 and NIH-R01-LM010101

** The author is partially supported by Helwan University, Cairo, Egypt.

and confidence ratio. An association rule is an implication $x \Rightarrow y$ where x and y are item-sets in a dataset. The *support* of the rule is the ratio of the number of transactions containing both x and y to the total number of transactions. The *confidence* of the rule is the ratio of the number of transactions that contain y to the number of transactions containing x . The mining of association rules from a set of transactions is the process of identifying all rules having a pre-specified minimum support and confidence. This involves several phases in processing transactions. A k -item-set is said to be *frequent* if the ratio of the number of transactions containing all the items of the k -item-set to the total number of transactions is greater than or equal to the user specified minimum support. The most time consuming part of rules mining is that of identifying all the frequent k -item-sets for all possible values of k . For this reason, papers published in the literature typically concentrate only on the problem of identifying frequent item-sets. Algorithms for finding frequent item-sets typically require multiple passes over the entire dataset. For example, the Apriori algorithm makes one pass through the data for each value of k [2].

Sampling can potentially make mining huge datasets feasible. Moreover, sampling algorithms may facilitate interactive mining [7]. When the goal is to obtain one or more interesting association rules as quickly as possible, a user might first mine a very small sample. If the results are unsatisfactory, the sample size can be iteratively increased until interesting rules are found. However, sampling may suffer from a couple of shortcomings. First is the presence of *missed item-sets* which are frequent in the entire dataset but infrequent in the sample. Second, there could be *false item-sets* which are infrequent in the entire dataset but frequent in the sample. These two issues could affect the accuracy of the sampling process.

The main contribution of this paper is ML-DS (Multi-Level Deterministic Sampling), a new deterministic sampling algorithm that attempts to improve accuracy without sacrificing the running time. ML-DS begins with a large sample deterministically selected from the dataset and then proceeds in levels. First, it divides the remaining data into disjoint groups of equal size. Each group in turn is recursively divided into smaller disjoint subgroups of equal size. A distance measure is then determined for each subgroup against the original group. Subgroups with minimum distance are retained while others are discarded. The process is repeated until the size of the remaining transactions is equal to a desired sampling threshold.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 describes the proposed algorithm ML-DS in detail. Experimental results are reported in Section 4. Section 5 concludes with some possible future work.

2 Related Work

The problem of discovering association rules consists mainly of two subproblems:

- Find all the item-sets that meet the pre-determined minimum support
- Use the frequent item-sets to generate the desired rules

While the rules generation process is straight forward, finding the frequent item-sets is a time consuming process. For this reason, research work on rules mining typically is focused on identifying frequent item-sets.

2.1 Mining Algorithms

There are many algorithms for mining association rules in the literature. Apriori [2] is possibly the well-known and most cited algorithm. It uses a breadth-first search strategy and generates candidate item-sets and tests if they are frequent. The key to its success over older algorithms such as AIS [1] and STEM [9] is the fact that it exploits an important property (commonly referred to as Apriori property or downward closure property). This property is the observation that no superset of an infrequent item-set can be frequent. However, generation of candidate item-sets is expensive both in space and time. In addition, support counting involves multiple dataset scans which heavily impact performance. Apriori as well as Apriori-inspired algorithms (e.g., [4]) typically perform well on sparse (i.e., short frequent item-sets) such as the market basket data. However, they perform poorly on dense (i.e., long frequent item-sets) datasets such as census data. This degradation is due to the fact that these algorithms perform as many passes over the database (i.e., high I/O overhead) as the length of the longest frequent pattern [17].

While many algorithms including Apriori use the traditional horizontal data layout, Eclat [19] is probably the first algorithm that uses a vertical data layout. Eclat is more efficient for long item sets than for short ones. In this algorithm, data is represented as lists of transactions identifiers (one per item). Support counting is performed by simply intersecting these lists. Compared to Apriori and other algorithms, Eclat often performs better on dense rather than sparse datasets. A variation of the algorithm depicted dEclat can be found in [17].

One more popular algorithm is the FP-Growth [8]. Unlike Apriori and other algorithms, it allows frequent item-sets discovery without candidate item-set generation and adopts a divide and conquer strategy. It builds a compact data structure (i.e., FP-Tree) using only two passes over the dataset. The algorithm is more efficient as long as the full tree can fit in memory. However, the tree may need substantial memory space in some cases. There are some variations of the algorithm such as H-mine [12].

TM algorithm [15] is a recent addition to the mining algorithms family. In this algorithm, transaction ids of each item-set are mapped and compressed to continuous transaction intervals in a different space and the counting of item-sets is performed by intersecting these interval lists in a depth-first order along the lexicographic tree. When the compression coefficient becomes smaller than the average number of comparisons for intervals intersection at a certain level, the algorithm switches to transaction id intersection. TM appears to rival older well established algorithms such as FP-Growth.

2.2 Sampling Algorithms

One of the oldest sampling algorithms for rules mining is by Toivnen [16]. The algorithm selects *randomly* a fixed size sample from the dataset. It finds all *maximal* and *negative border* item-sets in the sample. Note that an item-set is maximal if it is frequent but none of its supersets is frequent. An item-set is in the negative border if it is not deemed frequent in the sample but all its immediate subsets are. Upon checking with the remainder of the dataset (i.e., the dataset minus the sample), if no negative border item-sets turns out to be frequent, then the algorithm has succeeded and ends. Otherwise, another pass of the algorithm to eliminate false item-sets is required which can be quite expensive.

To tackle scalability issues of mining entire datasets, Zaki, et al. [18] argue that simple random sampling can reduce the I/O cost and computation time for association rule mining. The survey in [11] gives an overview of random sampling algorithms in databases. In [10], the authors propose two approaches to sample selection, namely, static sampling and dynamic sampling. With static sampling, a random sample is drawn from the large dataset, and hypothesis testing is used to establish whether it is sufficiently similar to the parent dataset. While in dynamic sampling, a decision is made after each sampled element whether to continue sampling or not. If the current sample is considered sufficient for the required level of accuracy, the sampling stops. Similarly, the authors of [13] propose using progressively larger samples as long as the model accuracy improves.

FAST [7] is another sampling algorithm. It proceeds in two phases. In the first phase, a large initial sample of transactions is selected *randomly* to estimate the support of each item in the dataset. In the second phase, these estimated supports are used to trim outlier transactions or select representative transactions from the initial sample, thereby forming a smaller final sample that potentially more accurately approximates item-set supports. EASE [5] is similar to FAST in that the final sample is obtained by deterministic methods from a larger random sample. EASE, however, uses an ϵ -approximation approach to obtain the final sample by a process of repeated halving. Our approach shares some ideas with these algorithms.

In [3], the authors propose two deterministic sampling algorithms, namely: Biased-L2 and DRS to find a sample from a dataset which optimizes the root mean square (RMS) error of the frequency vector of items over the sample (when compared to the original frequency vector of items in the entire dataset). The algorithms use ideas similar to EASE based on discrepancy theory [6], but sample the dataset without introducing halving rounds.

3 The Sampling Process

The ML-DS algorithm is inspired by the deterministic selection algorithm (DSelect) by Rajasekaran [14]. DSelect accepts a collection X of n keys and outputs the i th smallest key of the sequence. Before the sampling process of ML-DS is presented in details, the DSelect algorithm is briefly described in the following subsection.

```

1  $R_0 = X$ 
2  $j = 0$ 
3 while( $|R_j| > M$ )
4 {
5   divide  $R_j$  into disjoint groups of size =  $\min(M, |R_j|)$ ;
6   sort each group;
7    $j = j + 1$ ;
8    $R_j =$  keys whose ranks are  $(\sqrt{M}, 2\sqrt{M}, 3\sqrt{M}, \dots)$ 
9 }
10 return  $R_j$ 

```

Fig. 1. A stage of sampling

3.1 The DSelect Algorithm

The algorithm is reproduced in figure 2. Initially, all keys are considered *alive keys* (line 1). The algorithm goes through stages of sampling. A typical stage of sampling (line 7), expanded in figure 1, begins with dividing the collection $R_0 = X$ such that there are M keys in each part. After sorting each part, those keys that are at a distance of \sqrt{M} from each other are retained (i.e., it keeps the keys whose ranks are $(\sqrt{M}, 2\sqrt{M}, 3\sqrt{M}, \dots)$). Thus, the number of keys in the retained set R_1 is $\frac{n}{\sqrt{M}}$. Then, it groups the elements of R_1 such that there are M elements in each part, sorts each part, and retains only every \sqrt{M} th element in each part. Call the retained set R_2 . Proceed to obtain R_i 's in a similar fashion (for $i \geq 3$) until we reach a stage when $|R_j| \leq M$. If $n = M^c$, then clearly $j = 2c - 2$.

Once the sampling stage ends, two keys l_1 & l_2 are obtained that will bracket the key to be selected (line 8). A scan through the alive keys is conducted to kill keys that have a value outside the range $[l_1, l_2]$ (lines 11 and 12). Alive keys count is updated (line 14). The process is repeated until the number of the alive keys is less than or equal M . Remaining *alive keys* are sorted (line 16) and the i th smallest element is returned (line 17).

3.2 The ML-DS Algorithm

Formally, an association rule is an implication $x \Rightarrow y$ where x and y are *item-sets* in a dataset D . \mathcal{N} is the set of items in D where $N = |\mathcal{N}|$. The *support* s of the rule is the ratio of the number of transactions containing both x and y to the total number of transactions. The *confidence* c of the rule is the ratio of the number of transactions that contain y to the number of transactions containing x . \mathcal{S} denotes the sampling ratio and consequently the sampling threshold \mathcal{T} can be computed as follows:

$$\mathcal{T} = \mathcal{S} \cdot |D| / 100$$

$L_k(S)$ denotes the set of k -frequent item-sets from the sample S .

```

1 aliveKeys = X;
2 n = |aliveKeys|
3 repeat forever
4 {
5   if (n ≤ M)
6     break;
7   perform a stage of sampling for aliveKeys;
8   obtain 2 keys l1&l2 that will bracket the key to be selected;
9   foreach key k ∈ aliveKeys
10  {
11    if (k ∉ [l1, l2])
12      kill k;
13  }
14  n = |aliveKeys|;
15 }
16 sort aliveKeys;
17 return the ith smallest element;

```

Fig. 2. The DSelect Algorithm

In [7], the authors state that in the statistical literature, a well known technique for improving estimates obtained from a sample is to 'trim' the sample prior to computing the estimate. The idea is to make the sample more accurately reflect the properties of the parent population by removing 'outlier' observations that are not sufficiently representative of the data set as a whole. ML-DS employs these ideas and extracts a small final sample from a larger initial sample. Unlike most algorithms including [7] which use some randomization (e.g., select the initial sample randomly), ML-DS is purely deterministic.

The proposed algorithm ML-DS, depicted in figure 3, accepts a dataset of transactions D and outputs the mined frequent item-sets of a smaller sample S_0 extracted from D . ML-DS deterministically extracts an initial large sample S from D of size $|D|/2$ (line 1). The algorithm proceeds in levels until it reaches the pre-determined sampling threshold \mathcal{T} (line 4). At each level, the remaining data is divided into K groups G_i , $1 \leq i \leq K$, where $|G_i| = \alpha$ (line 6). In turn, each G_i is further divided into Z subgroups g_{ij} , $1 \leq j \leq Z$, where $|g_{ij}| = \beta$ (line 9). Observe that $\alpha = Z \cdot \beta$. For each subgroup g_{ij} , a distance measure d is determined against its parent group G_i (lines 10-11). d represents the difference between the supports of the 1-item-sets in g_{ij} and the corresponding supports in G_i . One way [7] to define d is as follows:

$$d(S_1, S_2) = \frac{|L_1(S_1) - L_1(S_2)| + |L_1(S_2) - L_1(S_1)|}{|L_1(S_1)| + |L_1(S_2)|}$$

where $S_1 \subset D$ & $S_2 \subset D$ and $L_1(S_1)$ & $L_1(S_2)$ denote the sets of frequent 1-item-sets in S_1 and S_2 , respectively. Note that d takes values in the range of (0-1) by definition. A value of 0 indicates minimum distance (i.e., exact frequent

item-sets). While a value of 1 indicates maximum distance (i.e., very different frequent item-sets).

```

1 extract deterministically an initial sample  $S$  from  $D$ ;
2 construct the items count matrix  $\mathcal{M}$ ;
3  $S_0 = S$ ;
4 while( $|S_0| > \mathcal{T}$ )
5 {
6   divide  $S_0$  into  $K$  disjoint groups of  $size = \min(\alpha, |S_0|)$ ;
7   foreach group  $G_i$ 
8   {
9     divide  $G_i$  into  $Z$  smaller disjoint subgroups of  $size = \min(\beta, |G_i|)$ ;
10    foreach subgroup  $g_{ij}$ 
11      compute distance  $d_{ij} = d(g_{ij}, G_i)$ ;
12      sort subgroups according to distances;
13       $S_0 = S_0 - \{G\}$  where  $\{G\} = \{g_{ij} : g_{ij} \in G_i \wedge j \geq Z/2\}$ ;
14    }
15 }
16 mine  $S_0$ 
17 return  $L(S_0)$ 

```

Fig. 3. The ML-DS Algorithm

The algorithm proceeds and sorts each subgroup $g_{ij} \in G_i$ based on the measured distance d (line 12). ML-DS retains only half the subgroups with minimum distances to their parent group and passes them to the next level. Remaining subgroups with higher distances are discarded (line 13). In the next level, the survived subgroups are allocated to larger groups of size α . The distance among subgroups and their parent groups are calculated and subgroups with minimum distances are retained while others are discarded. The same process is repeated until the required sampling threshold \mathcal{T} is obtained. Once the final sample S_0 is obtained, ML-DS mines all the frequent item-sets $L(S_0)$ using any established mining algorithm (line 16). We initially chose to use a publicly available implementation of Apriori by Borgelt¹ through out the algorithm. This implementation is widely used in academic and commercial purposes. However, the resultant implementation was very inefficient since there were many calls to this Apriori program. We have come up with a novel way to increase the efficiency. We only use Apriori to mine the final sample (line 16).

According to the distance formula, the frequent 1-item-sets L_1 for both large groups and their subgroups are repeatedly required. To increase the efficiency of this process, ML-DS begins with constructing a matrix of items count \mathcal{M} as the one in table 1. For each subgroup g_{ij} , the count of each item $i \in \mathcal{N}$ is recorded. Once this table is constructed, the frequent 1-item-sets for subgroups

¹ <http://www.borgelt.net/apriori.html>

can be retrieved in a constant time. For larger groups, a minimal computation is required to add the item count for each subgroup (e.g., G_1 in table 1 where $Z = 3$).

Table 1. Items Count Matrix

Item	g_{11}	g_{12}	g_{13}	G_1	g_{21}	g_{22}	g_{23}	G_2	...	g_{K1}	g_{K2}	g_{K3}	G_K
i_1	10	20	30	60	15	10	5	30	...	50	40	60	150
i_2	25	15	15	55	10	24	10	44	...	60	20	20	100
i_3	5	20	15	40	20	12	10	42	...	50	15	35	100
...
i_N	5	10	25	40	50	10	20	80	...	30	45	55	130

4 Experimental Results

This section demonstrates the potential of the proposed approach using several benchmarks. The experimental setup is first described followed by a discussion of both the capabilities and limitations of the approach.

4.1 Setup

Experiments were performed on synthetic datasets generated using the publicly available IBM quest synthetic data generator software². These datasets are shown in table 2 along with their parameters. The naming convention of the datasets follows the standards found in [2] which depicts some key parameters. The parameters include:

- T: the average length of a transaction
- I: the average length of the maximal potentially large item-sets (i.e., maximum pattern length)
- D: the number of transactions
- N: the number of items

Similar datasets were previously used in [7,3]. Reported results (both execution time and accuracy) are averages of 50 runs with a random shuffle of each dataset. Execution times are reported in seconds and include both the time to extract the sample *plus* the time to mine its frequent item-sets. The accuracy is calculated using this commonly used formula:

$$accuracy = 1 - \frac{|L(D) - L(S)| + |L(S) - L(D)|}{|L(S)| + |L(D)|}$$

where $L(D)$ and $L(S)$ denote all the frequent item-sets from the database D and the sample S , respectively. The sampling ratios used are 1%, 5%, 10%, and 15%. The minimum support ratios used are 0.5%, 0.75%, 1%, 1.5%, and 2%. All experiments were performed on a Windows 7 machine with processor speed at 2.80GHz and 3.0GB memory.

² <http://sourceforge.net/projects/ibmquestdatagen>

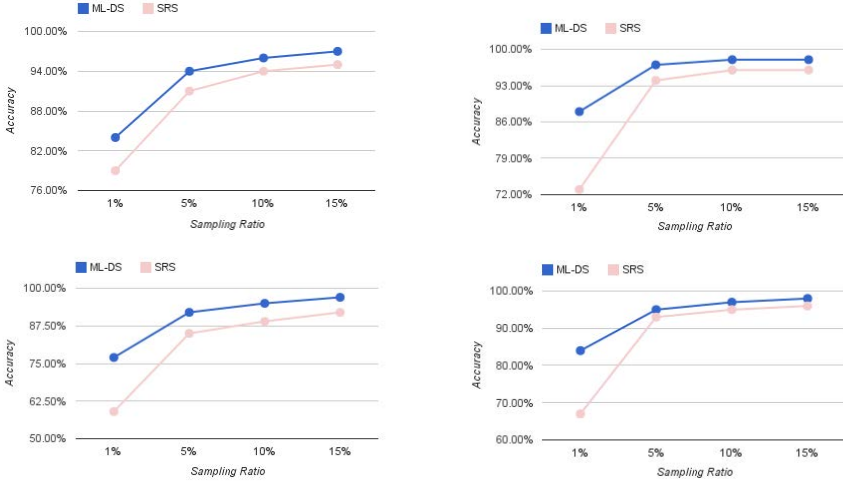


Fig. 4. Accuracy results (from top left) for datasets T5I3D100K, T5I10D100K, T10I4D100K, T10I6D100K with a minimum support of 0.75%

Table 2. Datasets & Parameters

Dataset	T(AvgTransLen.)	I(MaxPatternLen.)	D(TransCount)	N(ItemsCount)
T5I3D100K	5	3	100K	1000
T5I10D100K	5	10	100K	1000
T10I4D100K	10	4	100K	1000
T10I6D100K	10	6	100K	1000

4.2 Discussion

In this section, we discuss the performance of the proposed algorithm in terms of accuracy and execution time and show how it compares to Simple Randomized Sampling (SRS) algorithm and to reported results of a selected set of well established sampling algorithms as well.

Figure 4 illustrates the accuracy of both the SRS and ML-DS algorithms on the four datasets with a minimum support value of 0.75%. Clearly, ML-DS has a higher accuracy than that of SRS in every single case. For lower sampling ratios, the difference is more apparent with up to 20% improvement. For higher sampling ratios, however, the difference gap is lower. Similar results are obtained for other values of support.

Similarly, figure 5 shows the execution times of the two algorithms. Like all other sophisticated sampling algorithms, ML-DS is slower than SRS especially for smaller sampling ratios. However, the running time of ML-DS is increasing more slowly than that of SRS. In addition, the difference gap is expected to shrink as the sampling ratio gets bigger. Indeed, when the sampling ratio is bigger,

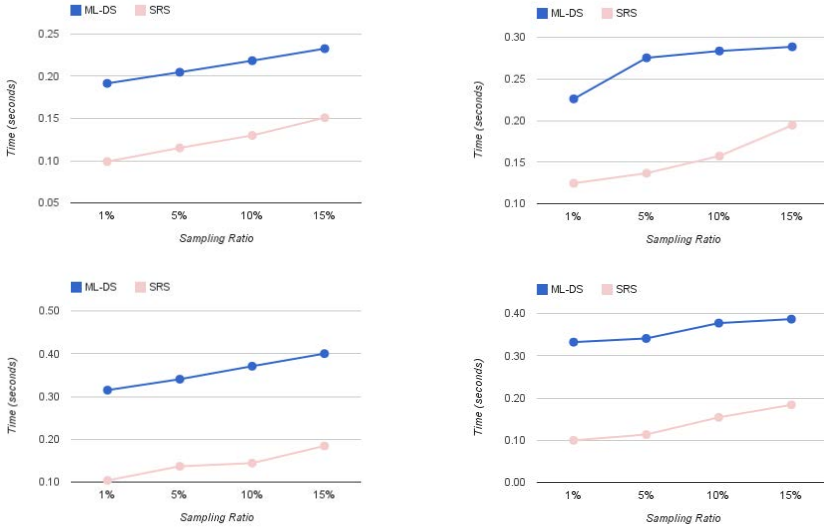


Fig. 5. Execution times (from top left) for datasets T5I3D100K, T5I10D100K, T10I4D100K, T10I6D100K with a minimum support of 0.75%

ML-DS performs less levels of sampling (i.e., less computation) while SRS does more work. Note that the time taken by ML-DS to extract and mine the sample is significantly smaller than the required time to mine the entire dataset.

In [7] the authors have shown that Toivonen's method is very accurate, but 10 times slower than FAST. Compared to reported results of FAST and other well established algorithms [7,5,3] on similar datasets, ML-DS is competitive in both running time and accuracy. ML-DS, however, has the advantage of being purely deterministic unlike most other algorithms. Moreover, due to the distributed nature of the algorithm, ML-DS can be easily adapted on a distributed system. In general, ML-DS is able to produce more stable results than SRS algorithm which gives fluctuating results (i.e., high standard deviation). In addition, the use of a simple random sample can lead to unsatisfactory results. The problem is that such a sample may not adequately represent the entire data set due to random fluctuation in the sampling process [5]. This difficulty is particularly apparent at small sample ratios.

The ML-DS has two influential parameters, namely, K and Z . The former dictates how many groups the data at each level is divided into. The later dictates how many subgroups large groups are divided into. We chose a value of 2 for both parameters since this was shown to be a reasonable choice in terms of both time and accuracy. The more groups in the process, the more will be the number of levels and consequently the execution time will be high.

5 Conclusions and Future Work

In this paper, we have introduced ML-DS, a novel deterministic algorithm for sampling large datasets. The algorithm begins with a large sample deterministically selected from the dataset. It then proceeds in levels and divides the data into large groups which subsequently get divided further into smaller subgroups. Only subgroups with minimum distances to its larger group qualify to the next level. The process is repeated until the desired sampling threshold is achieved.

Empirical results show that the approach outperforms simple randomized sampling in accuracy and is competitive in comparison with the state-of-the-art sampling algorithms in terms of both time and accuracy. Possible future research directions for this work include:

- Using different distance measures such as the ones found in [7].
- Optimizing the algorithm to improve the running time while keeping the accuracy as high as possible.

Acknowledgment. The first author is grateful to the the Computer Science Department, Helwan University, Cairo, Egypt for supporting his Ph.D. program at the University of Connecticut.

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, SIGMOD 1993, pp. 207–216. ACM, New York (1993)
2. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proc. 20th Int. Conf. Very Large Data Bases, VLDB 1994, vol. 1215, pp. 487–499 (1994)
3. Akcan, H., Astashyn, A., Brönnimann, H.: Deterministic algorithms for sampling count data. *Data Knowl. Eng.* 64, 405–418 (2008)
4. Bayardo Jr., R.J.: Efficiently mining long patterns from databases. In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, SIGMOD 1998, pp. 85–93. ACM, New York (1998)
5. Brönnimann, H., Chen, B., Dash, M., Haas, P., Scheuermann, P.: Efficient data reduction with ease. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2003, pp. 59–68. ACM, New York (2003)
6. Chazelle, B.: The Discrepancy Method. In: Chwa, K.-Y., Ibarra, O.H. (eds.) *ISAAC 1998*. LNCS, vol. 1533, pp. 1–3. Springer, Heidelberg (1998)
7. Chen, B., Haas, P., Scheuermann, P.: A new two-phase sampling based algorithm for discovering association rules. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2002, pp. 462–468. ACM, New York (2002)
8. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD 2000, pp. 1–12. ACM, New York (2000)

9. Houtsma, M., Swami, A.: Set-oriented mining of association rules. In: International Conference on Data Engineering (1993)
10. John, G., Langley, P.: Static versus dynamic sampling for data mining. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp. 367–370 (1996)
11. Olken, F., Rotem, D.: Random sampling from databases: a survey. *Statistics and Computing* 5(1), 25–42 (1995)
12. Pei, J., Han, J., Lu, H., Nishio, S., Tang, S., Yang, D.: H-mine: hyper-structure mining of frequent patterns in large databases. In: Proceedings IEEE International Conference on Data Mining, ICDM 2001, pp. 441–448 (2001)
13. Provost, F., Jensen, D., Oates, T.: Efficient progressive sampling. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 1999, pp. 23–32. ACM, New York (1999)
14. Rajasekaran, S.: Selection algorithms for parallel disk systems. *Journal of Parallel and Distributed Computing* 64(4), 536–544 (2001)
15. Song, M., Rajasekaran, S.: A transaction mapping algorithm for frequent itemsets mining. *IEEE Transactions on Knowledge and Data Engineering* 18, 472–481 (2006)
16. Toivonen, H.: Sampling large databases for association rules. In: Proceedings of the 22th International Conference on Very Large Data Bases, VLDB 1996, pp. 134–145. Morgan Kaufmann Publishers Inc., San Francisco (1996)
17. Zaki, M.J., Gouda, K.: Fast vertical mining using diffsets. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2003, pp. 326–335. ACM, New York (2003)
18. Zaki, M.J., Parthasarathy, S., Li, W., Ogihara, M.: Evaluation of sampling for data mining of association rules. In: Proceedings of the 7th International Workshop on Research Issues in Data Engineering, RIDE 1997, p. 42. IEEE Computer Society, Washington, DC (1997)
19. Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New algorithms for fast discovery of association rules. In: Knowledge Discovery and Data Mining, pp. 283–286 (1997)

Decision Rules Development Using Set of Generic Operations Approach

Wiesław Paja and Mariusz Wrzesień

Department of Artificial Intelligence and Expert Systems
University of Information Technology and Management in Rzeszów, Poland
{wpaja,mwrzesien}@wsiz.rzeszow.pl

Abstract. The main goal of presented research was to compile new approach for development learning models in a form of decision rule set. This approach devotes to using primary decision table as a primitive set of rules. Thus, each of learning cases is treated as a single classification rule. Next, a set of generic operations are applied to find the final, qualitative learning model. These generic operations are implemented in the RuleSEEKER system. During this research a few well-known algorithm for rule generation were compared with proposed solution. Obtained results are similar, sometimes even better and suggests that this method is a promising solution.

Keywords: classification rule, learning algorithms, rule optimization.

1 Introduction

One of the main task of data mining process is to assist users in extracting useful information or knowledge from the rapidly growing volumes of data. This knowledge is usually represented as a form of decision rule due to its easy understandability and interpretability. Classification rule mining are one of the major and traditional.

One reason why rules are popular is that each rule seems to represent some kind of an independent particle of knowledge. New rules can be added to an existing rule set without disturbing ones already there, whereas for example to add to a tree structure may require reshaping the whole tree. However, this independence is something of an illusion, because it ignores the question of how the rule set is executed. If rules are meant to be interpreted in order as a "decision list", some of them, taken individually and out of context, may be incorrect. On the other hand, if the order of interpretation is supposed to be immaterial, then it is not clear what to do when different rules lead to different conclusions for the same instance. This situation cannot arise for rules that are read directly off a decision tree because the redundancy included in the structure of the rules prevents any ambiguity in interpretation. But it does arise when rules are generated in other ways [12].

The problem of knowledge representation by means of decision rules is an important issue in many areas of machine learning domain. Decision rules have simple and understandable structure, however, in many practical applications even of quite trivial origin the number of rules in learning models can be disastrous. For this reason our research were devoted to the development of learning models displaying high efficiency and at the same time consisting of possibly low number of rules.

2 Theoretical Background

Inductive learning algorithms used commonly for development of sets of decision rules can cause the appearance of some specific anomalies in learning models [11]. This anomalies can be grouped as follows [8]:

- *redundancy*: identical rules, subsumed rules, equivalent rules, unusable rules,
- *consistency*: ambiguous rules, conflict rules, rules with logical inconsistency,
- *reduction*: reduction of rules, canonical reduction of rules, specific reduction of rules, elimination of unnecessary attributes,
- *completeness*: logical completeness, specific (physical) completeness, detection of incompleteness, and identification of missing rules.

These irregularities in learning models can be fixed (and sometimes removed) using some schemes generally known as verification and validation procedures [2]. Validation tries to establish the correctness of a system with respect to its use for a particular domain and environment. In short, we can agree that validation is interpreted as "building the right product", whereas verification as "building the product right". It has been argued that the latter is a prerequisite and subtask of the former. According to these mentioned anomalies it could be possible to optimize rule based learning models using some combination of generic operations.

3 Methods Used during Research

3.1 Generic Operations Algorithm

During our investigations a simple method based on generic operations was used. According to defined types of anomalies (see Section 2) which appear inside the gathered sets of rule a set of operations were implemented and tested. These operations were described in details in [9] and are the following:

- removing of redundant rules;
- removing of incorporative rules;
- merging rules;
- removing of unnecessary rules;
- removing of unnecessary conditions;
- creation of missing rules;
- selection of final set of rules.

Originally, this operations were implemented in RuleSEEKER system to perform an optimization of sets of rules gathered using well-known decision rule algorithms. The general structure of this system is presented on fig. 1. In this system, a set of developed rules is loaded simultaneously with learning decision table. Next, simple matching process is performed to find cases which are matched by particular rules. It enables to calculate parameters characterizing each of rules like strength, accuracy, support, specificity and generality. In the next step, a set of generic operations could be applied to the rule set. Some of them are based on mentioned parameters. Finally, system generates optimized set of rules according to error rate, quality of rules and their number. Here, only short description of an idea of RuleSEEKER system is presented.

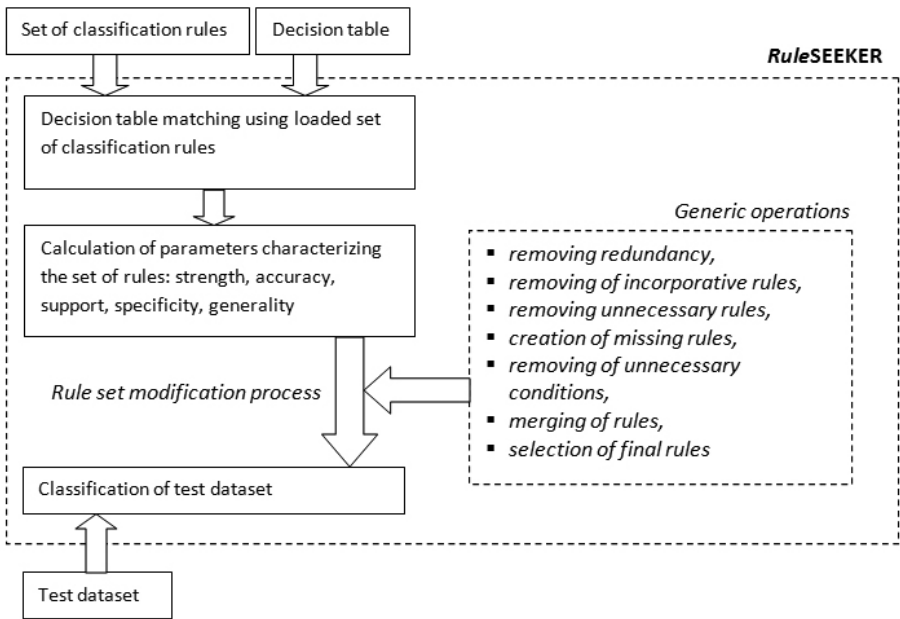


Fig. 1. General structure of the RuleSEEKER system

However, in this paper a new application is proposed. In this approach each learning case is treated as a single decision rule, thus generic operations are applied to primary dataset. In this way, by applying generic operations a new decision rule set is obtained. Finally, generated rules are evaluated by classification of test cases.

3.2 Other Rule Induction Algorithms

To assess results of application of generic operation approach to rule induction process three different rule induction algorithms were used. The first one, known

as GTS algorithm, belongs to general-to-specific group of algorithms and its functionality is based on sequential covering procedure. Details about this algorithm were presented in [75]. It is an inductive system starting from the most general rules and developing more specific decision rules in the learning process. To estimate quality of proposed condition following measure is used:

where: Generality = $(E_p + E_b) / E$ and Accuracy = $E_p / E_p + E_b$; E_p number of cases from learning data E that are correctly covered, E_b number of cases from learning data E that are incorrectly covered, E number of learning cases.

Next algorithm used in this research is the LEM2 algorithm introduced by Grzymala-Busse [4]. This algorithm is a local algorithm, dealing with attribute-value pairs, as opposed to global algorithms dealing with entire attributes. Basic notions used in LEM2 are blocks of attribute-value and decision-value pairs, minimal complexes, and coverings of concepts represented by decision-value pairs. As a result, algorithm LEM2 induces rules free from redundant attribute-value pairs. The third algorithm, called Tree-Via-Rule (TVR) [3], creates decision trees from fragments, which are sequences of paths from selected attributes to the decision attribute. In fact this is a set of induction rules.

3.3 Classification Process

In the classification process 10-fold cross-validation method was used. It is a statistical method of evaluating and comparing learning algorithms by dividing data into two segments: one used to learn or train a model and the other used to validate the model [10]. This operation has a few steps:

1. Break data into 10 sets of size $n/10$
2. Train on 9 datasets and test on 1
3. Repeat 10 times and take a mean accuracy

These stages are a well-known and standard scenario for performing an objective experiments. Additionally, during classification process the partial matching method were applied.

4 Investigated Datasets

Two different medical datasets were used during presented research. The first one concerns melanocytic skin lesion which is a very serious skin and lethal cancer. It is a disease of contemporary time, the number of melanoma cases is constantly increasing, due to, among other factors, sun exposure and a thinning layer of ozone over the Earth. Statistical details on this data are given in [6]. Descriptive attributes of the data were divided into four categories: *Asymmetry*, *Border*, *Color* and *Diversity of structures* (further called for short *Diversity*). The variable *Asymmetry* has three different values: *symmetric spot*, *one axial asymmetry* and *two axial asymmetry*. *Border* is a numerical attribute with values from 0

to 8. *Asymmetry* and *Border* are single-value attributes. The remaining two attributes, *Color* and *Diversity*, are multivalent attributes. *Color* has six possible values: *black*, *blue*, *dark brown*, *light brown*, *red* and *white*. Similarly, *Diversity* has five values: *pigment dots*, *pigment globules*, *pigments network*, *structureless areas* and *branched streaks*. Therefore, we introduced six single-valued variables describing color and five single-valued variables describing diversity of structure. In all of these 11 attributes the values are 0 or 1, 0 means lack of the corresponding property and 1 means the occurrence of the property. This dataset consists of 548 cases diagnosed by medical specialists using histopathological tests. All cases are assigned into four decision classes: *benign nevus*, *blue nevus*, *suspicious melanoma* and *melanoma malignant*.

The second dataset, with mental diseases cases, contains description of patients that were examined using the Minnesota Multiphasic Personality Inventory (MMPI) from the psychic disturbances perspective [1]. Examination results are presented in the form of profile. Patients profile is a data vector consisting of fourteen attributes. More exactly, a data vector consists of three parts:

- *Validity* part (validity scales): *lie*, *infrequency*, *correction*;
- *Clinical* part (clinical scales): *hypochondriasis*, *depression*, *hysteria*, *psychopathic deviate*, *masculinity-femininity*, *paranoia*, *psychasthenia*, *schizophrenia*, *hypomania*, *social introversion*;
- *Group* part a class to which the patient is classified.

Dataset consists of over 1700 cases classified by clinic psychologist. Each case is assigned to one of 20 classes. Each class corresponds to one of nosological type: *norm*, *neurosis*, *psychopathy*, *organic*, *schizophrenia*, *syndrome delusion*, *reactive psychosis*, *paranoia*, *manic state*, *criminality*, *alcoholism*, *drug induction*, *simulation*, *dissimulation*, *deviational answering style 1*, *deviational answering style 2*, *deviational answering style 3*, *deviational answering style 4*, *deviational answering style 5*, *deviational answering style 6*.

5 Obtained Results

Results of experiments are gathered in two tables. Each table contains name of algorithms used for rule induction, average number of rules inside developed learning models, and accuracy and error rate corresponding with investigated datasets. Table 1 consists results of analysis of melanocytic skin lesions dataset. It could be stressed that average number of rule in case of generic operations algorithm is equal to 114 and it is the lowest value. Additionally, error rate and accuracy for this solution are acceptable and even better than in case of GTS and LEM2 algorithms.

In turn, table 2 presents results of experiments gathered during analysis of the second dataset namely Minnesota Multiphasic Personality Inventory test results dataset. Also in this case generic operations approach supply good results. Average number of induction rules is equal to 396, and only the LEM2 algorithm has smaller value. However, average error rate value in this approach is equal to 30,28% and only TVR algorithm provide better result.

Table 1. Results of experiments for the first investigated dataset

Algorithm	Number of rules	Accuracy	Error rate
GTS	116	80,45%	19,55%
TVR	153	86,26%	13,74%
LEM2	119	68,08%	31,92%
Generic operations algorithm	114	83,83%	16,17%

Table 2. Results of experiments for the second investigated dataset

Algorithm	Number of rules	Accuracy	Error rate
GTS	508	67,20%	32,78%
TVR	444	80,01%	19,99%
LEM2	239	69,45%	30,55%
Generic operations algorithm	396	69,72%	30,28%

6 Conclusions

According to gathered results of experiments it could be admitted that proposed approach based on generic operations application on the set of rules is a promising solution. Presented results confirm good quantitative and qualitative estimation of developed learning models. Quantitative estimation is denoted by average number of developed rules. Qualitative estimation is denoted by average values of error rates and accuracy of learning models.

References

1. Gomuła, J., Pancerz, K., Szkoła, J.: Classification of MMPI Profiles of Patients with Mental Disorders – Experiments with Attribute Reduction and Extension. In: Yu, J., Greco, S., Lingras, P., Wang, G., Skowron, A. (eds.) RSKT 2010. LNCS, vol. 6401, pp. 411–418. Springer, Heidelberg (2010)
2. Gonzales, A., Barr, V.: Validation and verification of intelligent systems. *Journal of Experimental & Theoretical Artificial Intelligence* 12(2), 407–420 (2000)
3. Grzymała-Busse, J.W., Hippe, Z., Knap, M., Mroczek, T.: A new algorithm for generation of decision trees. In: Nowakowski, A. (ed.) *Computers in Medical Applications, Task Quarterly*, vol. 8, pp. 243–247. TASK Publishing, Gdańsk (2004)
4. Grzymała-Busse, J.: A new version of the rule induction system LERS. *Fundamenta Informaticae* 31, 27–39 (1997)
5. Hippe, Z.: Machine learning - a promising strategy for business information processing? In: Abramowicz, W. (ed.) *Business Information Systems 1997*, pp. 603–622. Academy of Economy Edit. Office, Poznan, Poland (1997)
6. Hippe, Z., Bajcar, S., Błajdo, P., Grzymała-Busse, J., Grzymała-Busse, J., Knap, M., Paja, W., Wrzesień, M.: Diagnosing skin melanoma: Current versus future directions. In: *Task Quarterly*, vol. 7, pp. 289–293. TASK Publishing, Gdansk (2003)
7. Hippe, Z., Hippe, T.: An attempt to automatize modeling of medical data. In: Kacki, E. (ed.) *Computers in Medicine*, pp. 24–31. Polish Society of Medical Informatics, Lodz, Poland (1997)

8. Ligeza, A.: Logical Foundations for Rule-Based Systems. Springer, Heidelberg (2006)
9. Paja, W., Hippe, Z.: Feasibility Studies of Quality of Knowledge Mined from Multiple Secondary Sources. I. Implementation of Generic Operations. In: Kłopotek, M., Wierzchoń, S., Trojanowski, K. (eds.) Intelligent Information Processing and Web Mining. AISC, vol. 31, pp. 461–465. Springer, Heidelberg (2005)
10. Refaeilzadeh, P., Tang, L., Liu, H.: Cross validation. In: Zsu, M.T., Liu, L. (eds.) Encyclopedia of Database Systems, pp. 27–39. Springer (2009)
11. Spreeuwenberg, S., Gerrits, R.: Requirements for successful verification in practice. In: Haller, S., Simmons, G. (eds.) Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference 2002. AAAI Press, Pensacola Beach (2002)
12. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Elsevier, San Francisco (2005)

Redundant Dictionary Spaces as a General Concept for the Analysis of Non-vectorial Data

Sebastian Klenk¹, Jürgen Dippon²,
Andre Burkovski¹, and Gunther Heidemann³

¹ Visualization and Interactive Systems Institute, Stuttgart University,
Stuttgart, 70569, Germany

klenksn@vis.uni-stuttgart.de

² Institute of Stochastics and Applications, Stuttgart University,
Stuttgart, 70569, Germany

³ University of Osnabrück,
Osnabrück, 49069, Germany

Abstract. Many types of data we are facing today are non-vectorial. But most of the analysis techniques are based on vector spaces and heavily depend on the underlying vector space properties. In order to apply such vector space techniques to non-vectorial data, so far only highly specialized methods have been suggested. We present a uniform and general approach to construct vector spaces from non-vectorial data. For this we develop a procedure to map each data element in a special kind of coordinate space which we call redundant dictionary space (RDS). The mapped vector space elements can be added, scaled and analyzed like vectors and thus allows any vector space analysis techniques to be used with any kind of data. The only requirement is the existence of a suitable inner product kernel.

Science [8,3] and social sciences [10] are generating non-vectorial data at a breathtaking pace, but still most analytical procedures require vectorial data. A great deal of research is therefore dedicated to the analysis of the various forms of non-vectorial data by transforming the original data into vectorial form. The transformation to a vector space allows the use of all vector space based methods, avoiding the need for new developments. The alternative is to develop completely new procedures and ignore existing research altogether. Some examples for this practice are Naive Bayes for classifying email spam [20] or customer sentiment [17], support vector machines to categorize text [7], hidden markov models for handling DNA sequences [12], and mixture models for network anomaly detection [6]. Further areas of non-vectorial data analysis are information retrieval, content filtering (spam, etc.), author identification, malware detection, biomedical data mining, multimedia and cross-media indexing, and text mining – each with its own approach. This proliferation of highly specialized procedures thwarts the development of new and innovative approaches to non-vectorial data analysis. Further, the lack of uniform and general treatment hinders a valid and thorough

comparison of existing approaches. Each new dataset requires the adaption of data processing and of analytical procedures. Therefore, the success or failure of a procedure can neither be attributed unequivocally to either the adaption or preprocessing of the data, nor to the procedure itself.

To carry out the analysis of non-vectorial data in the framework of vector spaces, the original non-vectorial data has to be transformed to a vectorial representation. These transformations are usually highly problem specific, so there is no common and uniform treatment for different kinds of non-vectorial data. But despite the effort of designing a suitable transformation, the treatment of the transformed data in the vector space is not straightforward: Basic vector operations such as calculating the norm, addition or scalar multiplication need to be modified. These modifications are necessary because even suitable transformations of non-vectorial to vectorial data are usually insufficient. In addition, a "hand-crafted" transform for a certain type of non-vectorial data usually cannot be applied to any other kind of non-vectorial data.

In this paper we show how to unify and generalize the treatment of non-vectorial data within vector spaces. In Section 1, we demonstrate what kind of errors are introduced when handling non vectorial data inappropriately. Based on these observations, in Section 2 we propose a general process to map non-vectorial data to a vector space. The resulting representations form a vector space in a strict mathematical sense, thus vector operations do no longer require any problem specific modifications. As a result, we can process all kinds of non-vectorial data in the same way as will be demonstrated in Section 3.

1 Handling Non-vectorial Data the Usual Way

Non-vectorial data is usually handled as a sequence of measurements (Figure 1 visualizes the process of the non-vectorial mapping). These measurements are, in general, highly redundant. Text, for example, is often represented as a vector of term occurrences where each term represents one coordinate and the value of that coordinate is the number of occurrences of that term within the text. Note that: (i) word occurrence vectors have an extremely high dimensionality where 40,000 dimensions are rather common, (ii) these vectors are very sparse with only very few non-zero coordinates and (iii) the word occurrences are highly correlated where the occurrence of one word makes the occurrence of others very likely. The correlation of coordinates introduces an error when calculating the norm or the inner product. To obtain an estimate of the size of such an error we make the following assumptions. First we assume that there is an underlying orthonormal data vector space $\mathcal{X} \subseteq \mathbf{R}^n$ that serves as a reference. We further assume that the measurements (μ^i) with $i \in \{1 \dots m\}$ are taken from the data space \mathcal{X} . We assume that there are at least as many measurements m as there are dimensions n with $m \geq n$ and we expect the measurement to, at least measure one dimension effectively, and non negative:

$$\mu_i^j \geq 0 \quad \text{and} \quad \mu_i^i = 1 \quad \text{for } i \leq n \text{ and } j \leq m.$$

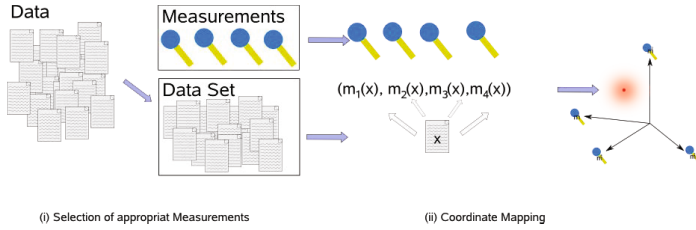


Fig. 1. The two steps (measurement selection and coordinate mapping) of the usual non-vectorial mapping process. The blurred point in the coordinate system represents the error introduced by the correlated measurements. These measurements conceal the exact position of the data element in the coordinate system.

We therefore obtain the value of a given measurement $\mu^i(x)$ for a data element x by calculating the inner product $\mu^i(x) = \langle \mu_i, x \rangle$. The vectorial representation of a data element x is therefore

$$v = (\mu^1(x), \dots, \mu^m(x)) = (\langle \mu^1, x \rangle, \dots, \langle \mu^m, x \rangle).$$

If we look at two data elements x^1, x^2 and their corresponding measurements v^1, v^2 , we observe that the measured distance between two data elements differs from the actual distance between the data elements by an additive term:

$$\begin{aligned} & \|v^1 - v^2\| - \|x^1 - x^2\| \\ &= \left(\sum_{i=1}^m (v_i^1 - v_i^2)^2 \right)^{1/2} - \|x^1 - x^2\| \\ &= \left(\sum_{i=1}^m \left(\sum_{j=1}^n \mu_j^i (x_j^1 - x_j^2) \right)^2 \right)^{1/2} - \|x^1 - x^2\| \\ &= \left(\sum_{i=1}^m \left((x_i^1 - x_i^2) + \sum_{j \neq i}^n \mu_j^i (x_j^1 - x_j^2) \right)^2 \right)^{1/2} - \|x^1 - x^2\| \\ &\geq \left(\sum_{i=1}^m \left((x_i^1 - x_i^2) + \sum_{j \neq i}^n \mu_j^i (x_j^1 - x_j^2) \right)^2 - \sum_{i=1}^m (x_i^1 - x_i^2)^2 \right)^{1/2} \\ &= \left(2 \cdot \sum_{i=1}^m (x_i^1 - x_i^2) \sum_{j \neq i}^n \mu_j^i (x_j^1 - x_j^2) + \sum_{i=1}^m \sum_{j \neq i}^n \mu_j^i (x_j^1 - x_j^2)^2 \right)^{1/2} \\ &\geq \left(2 \cdot \sum_{i=1}^n (x_i^1 - x_i^2) \sum_{j \neq i}^n \mu_j^i (x_j^1 - x_j^2) \right)^{1/2} \end{aligned}$$

The difference between the two distances depends on the correlation and is weighted by the difference of each of the data element coordinates. This term is

Table 1. Three terms and their eight most correlated terms with correlation. These terms are taken from the first 1,000 documents of the Reuters-21578 test collection. The table clearly shows that common phrases such as "and the", "new york" or "price per" are highly correlated. This correlation leads to an erroneous distance calculation when uncorrelated data is assumed.

Term								
	the	said	for	have	with	that	but	some
and	0.7500	0.7150	0.6135	0.5742	0.5712	0.5709	0.5400	0.5210
	york	the	for	and	said	venezuela	over	times
new	0.6418	0.3957	0.3820	0.3791	0.3315	0.3309	0.3203	0.3081
	day	per	ceiling	prices	spot	through	follows	but
price	0.2795	0.2752	0.2605	0.2585	0.2525	0.2519	0.2456	0.2449

an error term for it represents the differences between the actual distance and the measured distance.

Since the correlation between different measurements is a major source of inaccuracy, we will look at some examples to see the relevance of this observation for practical purposes.

Reuters NewsWire Articles. A common representation for text is counting term occurrences. We have taken the Reuters-21578 test collection [1] and calculated for each document the term frequency vector, i.e. the vector whose coordinates represent the terms in the document. The corresponding values are the number of occurrences of that term within the document. For the first 1,000 documents of the collection we calculated the correlation of the 1,000 most frequent terms. These correlations correspond to the measurement values μ_j^i in the calculations above. For demonstration we chose three intuitive and representative examples shown in Table 1. There is high correlation between the different words which leads directly to an increase of the error term in the equation above. The influence of this correlation becomes obvious for the terms "new" and "york". In many cases "new" is followed by "york" and a difference in "new" therefore means also a difference in "york". Such a difference is counted double.

Image Data. Another example for the negative influence of correlation between coordinates on vector space operations is given in the well known paper "Eigenfaces for Recognition" by Turk and Pentland [23]. The eigenvectors of the covariance matrix of a set of face images are used as a basis for a new and uncorrelated coordinate system. The result is a dramatic increase in recognition performance. In this example we have to be careful not to confuse a coordinate transformation that results in uncorrelated coordinates but still remains flawed by correlated measurements (as is the case for PCA) with an approach that changes the representation such that the measurements become uncorrelated (as for the "tightening" described below).

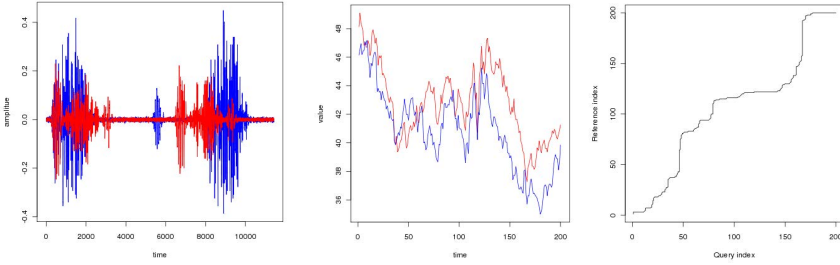


Fig. 2. Two examples of a phase and a frequency shift of events in time series. The first image to the right shows two heart sound that, due to different heart rates, have different time points at which the second heart beat occurs. Besides that, the blue curve (the one in the back) exhibits a third heart beat – the small one between the two large ones – that makes heart beat registration and normalization difficult. The two curves on the right show two stock market charts and their corresponding time wrapping function [21]. A time wrapping function is a transfer function which aligns two curves. In this case, it clearly shows two strong non linearities. These have to be considered when performing any vector operations on any of these curves.

Time Series Data. An other example of non vectorial data that introduces inaccuracies when being handled as vectorial is time series data. Here, not so much correlation, but varying semantics of vector coordinates pose the main difficulty. It is well known, that for most time series such as speech, biomedical or stock market data, variations of time points at which events occur or the speed at which events occur cause non linear fluctuations of the time axis [21]. These fluctuations can lead to a phase shift of a pattern, a frequency shift or both. Figure 2 visualizes these effects for two different data types. Fluctuations of the time axis require some form of registration, i.e. a alignment of each curve to certain characteristics. For heart beats this could be the registration of the two heart beats to certain fixed time points, for stock market data this could be the registration to an event common to all stocks such as the burst of a (stock market) bubble or the suspension of trade. But such an event is not necessarily given in all cases, which renders registration close to impossible. Figure 3 presents how such a case could be handled with a RDS, which we will introduce in the following section. As for all the examples above, the method proposed in the following section describes a way to overcome these difficulties and remove the introduced inaccuracies. The correlation of the coordinates is removed by the tightening and the registration problem is removed by using a kernel that aligns the data with each of the dictionary elements. This leads to aligned and uncorrelated coordinates. The resulting vectors can be handled like any vector without the risk of introducing errors through correlated coordinates or varying semantics of the coordinates.

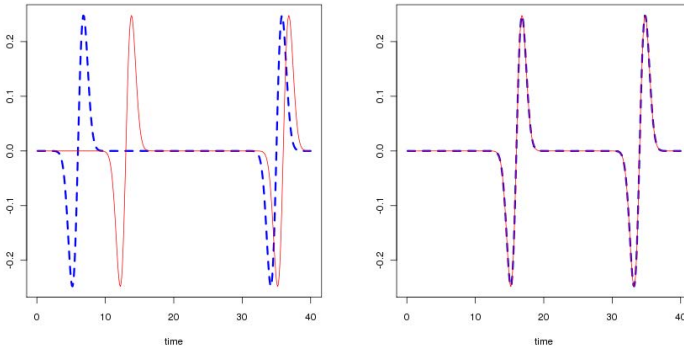


Fig. 3. The image on the left shows the abstraction of two different heart beats. Clearly visible is the difference in timing between the first and the second heart beat. If we perform a mapping in an RDS (and a reconstruction) we obtain two perfectly aligned heart beats (image on the right). This is because, when mapping the two heart beats in the RDS, we are using a dynamic time wrapping kernel that performs an alignment with the dictionary elements. Therefore, even though we have not specified any events that we want the time series to be aligned to, each time series is aligned to the dictionary elements and therefore every time series is aligned with every other.

2 The Redundant Dictionary Space

The general and uniform treatment of non-vectorial data as vector space elements is based on what we call *redundant dictionary spaces* (RDS), a coordinate vector space founded on a redundant dictionary. The process of mapping data in an RDS requires three steps: (i) The construction of a *redundant dictionary* [1](#), (ii) the *mapping* of the data elements in the coordinate space of the dictionary and (iii) the *tightening* of the coordinate vectors to remove interdependencies. Figure [4](#) visualizes the whole process. We will describe each of these steps in more detail later, but first we will take a look at the RDS and its mathematical foundation.

A RDS \mathcal{R} is a coordinate vector space whose coordinates are the coefficients of linear combinations of dictionary elements $(d^i)_{i \in \{1 \dots n\}} = \mathcal{D}$ [2](#). These linear combinations are elements of the data space \mathcal{X} from which also the dictionary elements are taken.

$$v \in \mathcal{R}, d^i \in \mathcal{D} \subset \mathcal{X} \quad \text{and} \quad x = \sum_i v_i d^i \in \mathcal{X}$$

¹ Any sequence of unit vectors that spans part of the vector space (or the entire vector space) is called a dictionary, regardless of linear dependence. Thus dictionaries may be highly redundant. By contrast, a basis is a minimal set of vectors spanning the entire vector space.

² Up to the next section will consider the data and the dictionary elements to be elements of an ordinary vector space. Later on we will demonstrate that we can loosen this requirement such that dictionary and data may be non-vectorial.

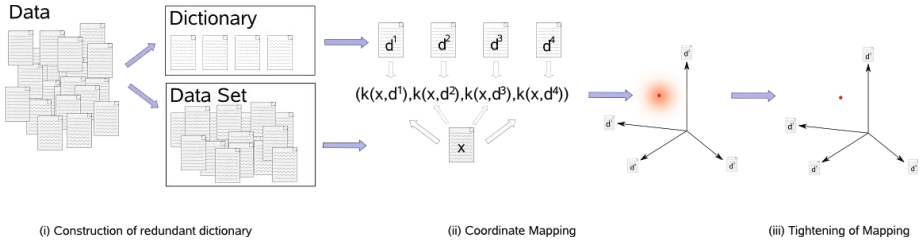


Fig. 4. The three steps (dictionary selection, mapping and tightening) of the redundant dictionary mapping process. The blurred point in the first coordinate system represents the error introduced by the redundant dictionary elements. These redundancy within the coordinates is removed by the tightening step which leads to an exact position in the second coordinates system.

Because of the redundancy in the dictionary, component wise different RDS elements can form the same vector space element

$$u, v \in \mathcal{R} \text{ with } v_i \neq u_i \text{ for some } i \in \{1 \dots n\} \text{ and } x = \sum_i v_i d^i = \sum_i u_i d^i.$$

The RDS elements u and v can therefore be considered equivalent. A RDS is the quotient space $\mathcal{R} = R^n / \sim$ of a coordinate space of dimension $n = |\mathcal{D}|$. The equivalence relation of this quotient space is given by the equivalence in the data space

$$u \sim v \text{ iff } \sum_i v_i d^i = \sum_i u_i d^i.$$

As a quotient space, an RDS is a vector space and can be handled as such [19].

Our main goal, when introducing the RDS was to provide a more comfortable way to handle elements of the data space \mathcal{X} spanned by the dictionary. This means that operations on RDS elements should yield identical results as those performed on the corresponding data space elements. These operations include the vector space operations addition, scalar multiplication and also the calculation of the inner product. Therefore we introduce the redundant dictionary inner product (RDIP, a function that maps two RDS elements in \mathbf{R}). It is constructed such that the interdependence between the dictionary elements is taken into account:

$$\langle u, v \rangle_{\mathcal{R}} = \sum_i \sum_j \lambda_{ij} u_i v_j \text{ with } \lambda_{ij} = k(d^i, d^j) \sim \langle d^i, d^j \rangle$$

A more comfortable way to write the inner product is with the Gramian matrix $G = (\langle d^i, d^j \rangle)_{i,j \in \{1 \dots n\}}$ of the dictionary: $\langle u, v \rangle_{\mathcal{R}} = u^T \cdot G \cdot v$. The RDS \mathcal{R} – a quotient space given by the equivalence class of coefficient vectors resulting in the same dictionary element over the vector space R^n – and the RDIP $\langle \cdot, \cdot \rangle_{\mathcal{R}}$ form a vector space with inner product. Within the RDS we can treat elements of \mathcal{X} as if they were vectorial and, depending on the approximative quality of the mapping, we can expect the results to be similar to those performed (if they were defined) in the original space. This fact is especially interesting if the data space is non-vectorial.

3 The Redundant Dictionary Mapping Process

The cornerstone of the general and uniform treatment of non-vectorial data described in this paper is the redundant dictionary. The first step of the mapping process is the selection of the dictionary elements. These are sampled randomly from the dataset and are therefore of the same data type as the data elements and usually highly redundant. The dictionary defines the mapping in the RDS, as well as the tightening step. An overview of all necessary steps of the redundant dictionary mapping process can be found in Table 3.

3.1 Mapping

Usually, if we are mapping vectorial data in a coordinate space of a basis $\mathcal{B} = (b^1, b^2, \dots)$, we calculate the inner product of each data element with each basis element. This results in as many coefficient vectors as there are data elements and each coefficient vector consists of as many coordinates as there are basis elements.

$$(v_1, v_2, \dots) = (\langle x, b^1 \rangle, \langle x, b^2 \rangle, \dots) \quad (1)$$

If the basis is orthonormal, this mapping allows a direct reconstruction of the original data elements as a linear combination of the basis elements given the coordinates of the coefficient vector: $x = \sum v_i b^i$. In case of a redundant dictionary the mapping is handled the same way: for each data element we calculate the inner product between the data element and each dictionary element. This yields a coefficient vector with as many coordinates as there are dictionary elements. Similar to the mapping on a basis the mapping on a dictionary takes as input a data space element and outputs a coordinate vector. But unlike the mapping on a basis, a mapping to a dictionary does not allow the reconstruction of the data element. We therefore have to take one further step, the tightening described in the following section.

3.2 Tightening

Due to the redundancy in the dictionary – it is neither necessarily orthogonal nor linearly independent – a mapping by an inner product results in a coefficient vector that is loaded with interdependencies between the coefficients. For example, a mapping in the coordinate space of a dictionary which contains two identical elements would result in two identical coefficients, each corresponding to the contribution of one of the identical dictionary elements. This dictionary element is therefore represented twice. This implies that the mapping of a data element in the coordinate space of a dictionary does not allow reconstruction. Further, addition and scalar multiplication cannot yield results similar to those performed in the data space. Besides, an inner product would also be based on the coefficients of the incorrect representation and would weight the influence of redundant dictionary elements multiple times. We therefore have to remove these interdependencies to obtain a representation that allows an exact reconstruction

Table 2. The matching pursuit algorithm with adaption to the non-vectorial case. The input to the algorithm are the RDS mapping $v = (k(x, d_1), k(x, d_2), k(x, d_3), \dots, k(x, d^m))$ of the data element x , the Gramian $G = (\lambda_{ij}) := \nu_i \nu_j k(d_i, d_j)$ of the normalized dictionary \mathcal{D} and the number k of expected non zeros and the acceptable error ϵ .

Steps Matching Pursuit		
1	Initialize	Initialize the output vector $v' = (0, 0, \dots)$ with zeros.
2	Best Match	Find i such that $i = \operatorname{argmax}_{\in\{1\dots m\}} v_j $.
3	Update	Set $v'_i = v_i$, $v_i = 0$ and for all $j \in \{1 \dots m\}, j \neq i$ set $v_j = v_j - v_i \cdot G_{ij}$.
4	Iterate	Repeat 2 and 3 while the number of non zeros in v' is less than k and $ v_i \geq \epsilon$.

of the data element and calculations that correspond to those performed in the data space. One way to remove these dependencies is with the help of Mallats matching pursuit [14]. This is a method to calculate sparse coefficient vectors of data elements given an overcomplete dictionary. The details of the algorithm are presented in Table 2.

3.3 Inner Product Kernel

Mapping data elements in an RDS requires the calculation of an inner product. However, if the data elements are non-vectorial, there is no inner product defined on them. But usually there are so called kernel functions defined on the data.

The term Kernel derives from the theory of integral equations. Kernels are functions of two variables and are square summable with respect to each variable. They are used in integral operators to transform functions from one domain to another. In the context of machine learning, special kernels – commonly referred to as inner product or mercer kernels – are used to calculate the inner product of the two input variables in some high dimensional vector space. Necessary and sufficient conditions for a kernel to be an inner product kernel are stated in the Mercer Theorem [24].

Inner product kernels are not limited to vectorial data. There are numerous kernels defined on non-vectorial data [5], examples are string kernels [13] for text data, graph kernels [26] for graph or tree like structures and local alignment kernels for biological sequences [25]. These are functions that take as an input two data space elements and output a real number. They allow the calculation of an inner product of two non-vectorial variables. The non-vectorial data of these two variables is thereby transformed into vectorial form by some (abstract) transformation Φ .

$$(v_1, v_2, \dots) = (\langle \Phi(x), \Phi(b^1) \rangle, \langle \Phi(x), \Phi(b^2) \rangle, \dots) = (k(x, b^1), k(x, b^2), \dots)$$

This way, an inner product can be calculated on non-vectorial data and, with the help of such a kernel function, we can calculate a mapping of non-vectorial data on an RDS.

Table 3. The RDS mapping and tightening of a dataset $\mathcal{X} = \{x^1, \dots, x^n\}$ for further use in vector space methods

Steps	RDS Mapping	
1	Selection of dictionary	Randomly select m dictionary elements $\mathcal{D} = \{d^1, d^2, \dots, d^m\}$ from the dataset and calculate a normalization constant $\nu_i = 1/\sqrt{k(d^i, d^i)}$ for each dictionary element.
2	Mapping in RDS	Calculate for each data element x its mapping in the RDS $x \rightarrow v = (\nu_1 \cdot k(x, d^1), \nu_1 \cdot k(x, d^2), \dots, \nu_1 \cdot k(x, d^m))$.
3	Tightening in RDS	Tighten the redundant representation v by calculating its matching pursuit representation v' in the RDS.
4	Vector space calculation	Now the RDS elements v' can be used instead of the data space elements to perform vector space calculations.

4 Numerical Experiments

Traditionally, using vector space methods on non-vectorial data raises lots of difficulties. The process proposed in this paper reduces this treatment to the selection of a suitable kernel so different data types can be handled identical, with one general procedure. It should be noted that the procedure applies to all data types for which an inner product kernel is defined. For demonstration purposes we will cluster each dataset with a self organizing map (SOM) [11], an artificial neural network that makes heavy use of the different vector space properties [3]. The choice of SOMs as an example is arbitrary – any method based on vectorial data would work. We have chosen SOMs because they exploit all vector space properties and are widely used [9].

All calculations are based on the plain vanilla version of the SOM algorithm and are exactly as described by Kohonen [11] for vectorial data. The only exception is the calculation of the norm $\|v\| = \langle v, v \rangle_{\mathcal{R}}^{1/2}$,

4.1 Reuters Newswire Articles

The first example is the clustering of text data. For this purpose we use the Reuters-21578 [1] test collection, a set of 21,578 Reuters Newswire texts that are categorized by human experts into 120 topics. In this experiment we will group the articles with a SOM and observe how the grouping corresponds to the assigned categories. For our experiment we employ the modified Apte split (ModApte) as described in [1], consisting of 9,603 training documents, as data basis. From this dataset we randomly sample 40 dictionary elements. The number of elements can be obtained by calculating the expected approximation error of

³ Materials and methods are available as supporting material on the authors homepage <http://www.vis.uni-stuttgart.de/institut/mitarbeiterinnen/klenksn.html>

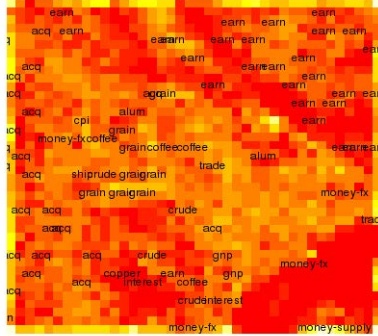


Fig. 5. The heat map representation of the SOM, fit to the 9,603 documents of the ModApte-Split of the Reuters 21578 test collection. The color coding represents the density of the nodes, the lighter the color the higher the density. The labels of the nodes are the dominant topics of the assigned documents.

the mapping as shown in Figure 8. We map the remaining 9,563 documents in the RDS with a String Kernel [13] and use these resulting 9,563 RDS elements to train a 40×40 2D SOM. We train the SOM over 100 iterations and initialize the nodes with randomly sampled data elements. The training rate $\alpha = 0.1$ is fixed and we use $\sigma = 4$ as proximity factor in the gaussian density function h . For each document, after training, calculate the best matching node in the SOM and observe how the assigned node corresponds to the topic of the document. The results of the calculation can be seen in Figure 5. From the node assignment we calculated precision (0.8925, 0.6310, 0.6529), recall (0.8966, 0.7553, 0.5675) and F_1 (0.8946, 0.6876, 0.6072) scores for the three most frequent topics "earn", "acq" and "money-fx". The numbers indicate that the majority of the data elements with the topic "earn" are assigned to nodes with topic "earn" and that most nodes with this topic contain only data elements with the corresponding topic, i.e. the SOM largely assigns the data elements to the correct nodes. For the other topics there is still a large number of data elements assigned correctly but there is also some overlap with other topics.

To compare the results with existing approaches we clustered the same data, this time in form of term occurrence vectors, with the same SOM algorithm. Each data element was encoded as 112⁴ terms occurrence vectors which we obtained by transforming all characters to lower case, removing all stop words, removing all number and punctuation and selecting the most frequent terms. The calculations resulted in the following scores: precision (0.7064, 0.6741, 0.5540), recall (0.8960, 0.6415, 0.3347) and F_1 (0.7900, 0.6574, 0.4173). As the numbers indicate, the RDS approach yields better results. When comparing these numbers with existing classification results [7] we have to consider that the proposed

⁴ We selected the most frequent terms with the R Text Mining Package 'tm' [4] where 112 terms are selected by sparsity.

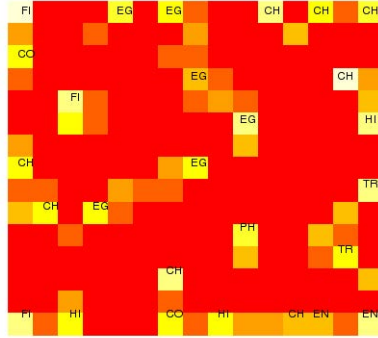


Fig. 6. The heat map representation of the 2D SOM, fit to the DAX stocks. The labels represent the eight industries (EnGineering, FInance, TRansportation and COmmunication, CHEmical, PHarmaceutical, MEdical, COmmerce, and HIgh Tec) that are represented in the index. The SOM clearly shows the clusters of main industries such as Transportation, Chemistry and Engineering. Other clusters are more interwoven and do not allow a discrete grouping, this is also due to companies that span multiple industries such as Siemens or Daimler or some financial institutions.

approach is not a task specific classification method but a generally applicable method.

4.2 Stock Market Data

Our second example are stock market data. Even though this kind of data looks like vectorial data, it can hardly be treated in such a way. Stock market charts are time series with all the requirements for normalization and registration [18], so a high degree of preprocessing is required. A suitable kernel though is capable of incorporating all these preprocessing steps and allows to map the raw data to the dictionary. As dataset we use the charts of the stocks of the german stock index DAX from 01.01.2010 to 01.08.2011, downloaded from the Yahoo Finance website [2]. To increase the number of data elements, we split the 30 time series into sequences of 50 values which elevates the number of data elements to 237. From this set we select 15 dictionary elements and we map the remaining 222 data elements in the RDS with a dynamic time wrap (DTW) [21] based kernel. This kernel calculates the time normalized distance between two time series and uses a gaussian function to obtain similarities. Similar to the Reuters Newswire example above, we use a a 2D 15 × 15 SOM and training parameters $\alpha = 0.5$ and $\sigma = \sqrt{2}$. The algorithm is identical to that used for the text data. The resulting SOM can be seen in Figure 6.

4.3 Image Data

In our last example we cluster image color histograms. Color histograms are vectorial data, nonetheless the approach presented in this paper can be applied the

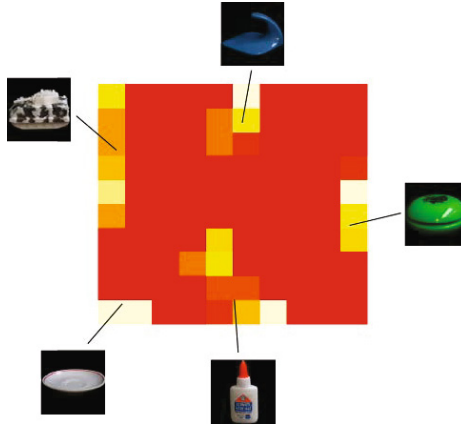


Fig. 7. The heat map representation of the $2D$ SOM with 10×10 nodes, fit to the color histograms of 320 images taken from the COIL Image Database. Each square represents one node, the color coding represents the density of the node, i.e. the number of data elements that match the corresponding weight vector is higher for light colored nodes than for darker ones. The labels are assigned to those nodes with the highest density in a given cluster. The label is that of the dominant class of the assigned data elements.

same way as described above. As dataset we use images taken from the COIL Image Database [16]. The images show four different objects in different positions. From each image we calculate the color histogram (a vector of 36 values, each corresponding to 10 degrees in the HSV color cone). These histograms are plain vectorial and we will show that the process presented here actually leads to identical results as if plain vectorial algorithms were used [15]. The dataset consists of 360 histograms (72 for each of the 5 elements). We are randomly selecting 17 dictionary elements and are using the ordinary inner product to map the histograms in the RDS. We use a 10×10 $2D$ SOM with parameters $\alpha = 0.01$ and $\sigma = \sqrt{2}$. Again, the algorithm is identical to that used in the other examples. The results can be seen in Figure 7. The grouping of the objects is, as in Moehrmann et al., exact. This means that each node contains only histograms of one of the displayed objects and nodes with histograms of identical objects form distinct clusters.

5 Conclusion

In this paper we have presented a general method to handle non-vectorial data in a vector space. We have demonstrated how any kind of data, for which an inner product kernel is defined, can be mapped in a vector space. All operations defined on vector spaces can be applied to the vectorial representation and we can expect to obtain similar results as those performed on the original data. The treatment and analysis of non-vectorial data is an important task. Non-vectorial data is generated in numerous scientific areas. To our best knowledge, there

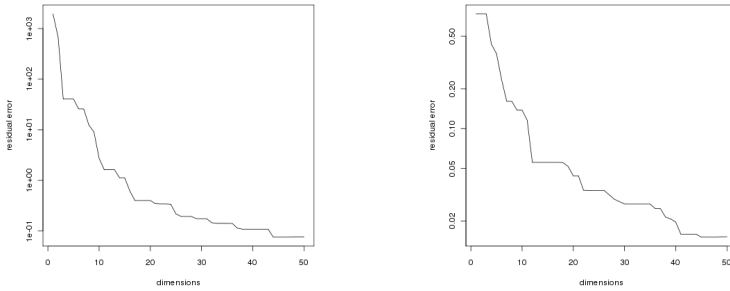


Fig. 8. The estimated residual error in log-scale of the mapping with increasing dimension of two datasets. On the left, the residual error of the color histogram dataset and on the right, the residual error of the finance dataset. The most suitable number of dictionary elements for the mapping can be found by looking at the "elbow" at which the decrease of the values slows down. This procedure is commonly employed to estimate the most suitable dimension in dimensionality reduction problems such as PCA, MDS or IsoMap [22]. The red line marks the corresponding position.

is currently no general process to handle non-vectorial data in a general and unified way. Depending on the data type, specially crafted methods have to be employed. With the help of the approach presented in this paper, different data types can be handled with identical methods and even with an identical software code base. Research benefits from the unified treatment of data and the resulting comparability of results. With identical data handling, preprocessing and data representation can be ruled out as a source for different results. Therefore the focus of the analysis can be placed on the method and on the experiments, instead of the data representation. Application design benefits from the use of one single code base for analytical procedures. The code can be easily extended to new data types: The only thing that has to be changed is the kernel. The process and the examples we have presented are very promising. There are many different areas of application, but there is also need for further research. So far, we have not investigated the influence of the kernel on the mapping. The assumptions in this paper are based on the mathematical idea of a kernel function. The definition of such a function is rather broad and allows many different kernels for a given data type. Therefore, further research is necessary to actually determine suitable kernels for different data types.

References

1. The Reuters-21578, Distribution 1.0 test collection, <http://www.daviddlewis.com/resources/testcollections/reuters21578>. We are using the XML-encoded version of Reuters-21578 from Saturnino Luz, <http://modnlp.berlios.de/reuters21578.html>
2. Yahoo Finance, <http://finance.yahoo.com/>, is a website that provides programmatic access to financial data. The web service is documented in, <http://code.google.com/p/yahoo-finance-managed/wiki/YahooFinanceAPIs>

3. Akil, H., Martone, M.E., Van Essen, D.C.: Challenges and opportunities in mining neuroscience data. *Science* 331, 708–712 (2011)
4. Feinerer, I., Hornik, K., Meyer, D.: Text mining infrastructure in R. *J. Statistical Software* 25 (2008)
5. Gärtner, T.: A survey of kernels for structured data. *SIGKDD Explor. Newsl.* 5, 49–58 (2003)
6. Hajji, H.: Statistical analysis of network traffic for adaptive faults detection. *IEEE Trans. Neural Networks* 16(5), 1053–1063 (2005)
7. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: *Eur. Conf. Mach. Learn. (ECML)*, pp. 137–142. Springer, Berlin (1998)
8. Kahn, S.D.: On the future of genomic data. *Science* 331, 728–729 (2011)
9. Kaski, S., Kangas, J., Kohonen, T.: Bibliography of self-organizing map (SOM) papers: 1981–1997. *Neural Comput. Surv.* 1, 102–350 (1998)
10. King, G.: Ensuring the data rich future of the social sciences. *Science* 331, 719–721 (2011)
11. Kohonen, T.: *Self-organizing maps*, 3rd edn. Springer, Berlin (2001)
12. Krogh, A., Brown, M., Saira Mian, I., Sjander, K., Haussler, D.: Hidden markov models in computational biology: applications to protein modeling. *J. Mol. Biol.* 235, 1501–1531 (1994)
13. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: Text classification using string kernels. *J. Mach. Learn. Res.* 2, 419–444 (2002)
14. Mallat, S.G., Zhang, Z.: Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.* 41(12), 3397–3415 (1993)
15. Moehrmann, J., Bernstein, S., Schlegel, T., Werner, G., Heidemann, G.: Improving the Usability of Hierarchical Representations for Interactively Labeling Large Image Data Sets. In: Jacko, J.A. (ed.) *HCI International 2011, Part I. LNCS*, vol. 6761, pp. 618–627. Springer, Heidelberg (2011)
16. Nayar, Murase, H.: Columbia object image library: COIL-100. Technical Report CUCS-006-96, Department of Computer Science, Columbia University (February 1996)
17. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: *Proc. Conf. Emp. Meth. Nat. Lang. Proc. EMNLP*, pp. 79–86 (2002)
18. Ramsay, J.O., Silverman, B.W.: *Functional Data Analysis*. Springer, Berlin (2005)
19. Rudin, W.: *Functional analysis*, 2nd edn. McGraw-Hill, Boston (1991)
20. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian approach to filtering junk e-mail. In: *Proc. AAAI 1998 Workshop on Learn. Text Cat.* (1998)
21. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Signal Process.* 26(1), 43–49 (1978)
22. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 2319–2323 (2000)
23. Turk, M., Pentland, A.: Eigenfaces for recognition. *J. Cognitive Neuroscience* 3, 71–86 (1991)
24. Vapnik, V.N.: *Statistical learning theory*. Wiley, New York (1998)
25. Vert, J.-P., Saigo, H., Akutsu, T.: Local Alignment Kernels for Biological Sequences, pp. 131–153. MIT Press, Cambridge (2004)
26. Vishwanathan, S.V.N., Schraudolph, N.N., Kondor, R.I., Borgwardt, K.M.: Graph kernels. *J. Mach. Learn. Res.* 11, 1201–1242 (2010)

Human-Centered Text Mining: A New Software System

Jonas Poelmans^{1,5}, Paul Elzinga³, Alexei A. Neznanov⁵, Guido Dedene^{1,4},
Stijn Viaene^{1,2}, and Sergei O. Kuznetsov⁵

¹ KU Leuven, Faculty of Business and Economics, Naamsestraat 69,
3000 Leuven, Belgium

² Vlerick Leuven Gent Management School, Vlamingenstraat 83,
3000 Leuven, Belgium

³ Amsterdam-Amstelland Police, James Wattstraat 84,
1000 CG Amsterdam, The Netherlands

⁴ Universiteit van Amsterdam Business School, Roetersstraat 11
1018 WB Amsterdam, The Netherlands

⁵ National Research University Higher School of Economics (HSE), Moscow, Russia
{Jonas.Poelmans, Stijn.Viaene, Guido.Dedene}@econ.kuleuven.be,
{aneznanov, skuznetsov}@hse.ru,
Paul.Elzinga@amsterdam.politie.nl

Abstract. In this paper we introduce a novel human-centered data mining software system which was designed to gain intelligence from unstructured textual data. The architecture takes its roots in several case studies which were a collaboration between the Amsterdam-Amstelland Police, GasthuisZusters Antwerpen (GZA) hospitals and KU Leuven. It is currently being implemented by bachelor and master students of Moscow Higher School of Economics. At the core of the system are concept lattices which can be used to interactively explore the data. They are combined with several other complementary statistical data analysis techniques such as Emergent Self Organizing Maps and Hidden Markov Models.

Keywords: Formal Concept Analysis, Text Mining, Software System, Applications, Concept lattices.

1 Introduction

A crucial enabler for innovation in 21st century data-intensive organizations is being able to deal with massive amounts of textual information. Amongst others in the crime data mining framework of Chen et al. (2004) text mining was pointed out as a promising research field. Unfortunately till date only few successful applications have been reported on in the literature. Over the past years several papers have been published on applying Natural Language Processing (NLP) techniques and extracting key words from texts often with the aim of building an ontology and classifying some documents (e.g. Cimiano et al. 2005, Maio et al. 2012). In this paper we try to go further and focus on semi-automatically exploring textual data using visual models. More in particular we showcase the “COncept Relation Discovery and Innovation

Enabling Technology” (CORDIET) software system which takes its roots in several real-life case studies with the Amsterdam-Amstelland Police and GZA hospitals.

Before our research, no automated analyses were performed on the observational reports written by officers and filed in the Amsterdam-Amstelland police region. The reason was an absence of good instruments to detect the observations containing interesting information and to analyze the texts they contain. Only on the structured information stored in police databases, analyses were performed. These include the creation of management summaries using Cognos information cubes, geographical analysis of incidents with Polstat and data mining with Datadetective. A few projects were devoted to automatically identifying domestic violence in statements made by victims however the results were not convincing enough to make it into operational policing practice.

At the core of our system are concept lattices (Wille 1982, Ganter et al. 1999) which can be used to visualize and interactively gain insight in the underlying concepts of the data. The lattice-based analysis can be combined with Hidden Markov Models (HMM) (Rabiner 1989) and Emergent Self Organising Maps (ESOM) (Ultsch 2003, Ultsch 2005). We chose for a human-centered (Fayyad 2002) setup of the system. A domain expert, who is not a trained computer scientist or statistician, can apply a powerful arsenal of analysis methods to his particular problem and adapt them to his needs without having to deal constantly with technical details.

The remainder of this paper is composed as follows. Section 2 describes the setup of the project and the software system. In section 3, an overview of the datasets used during the research is given. In section 4 we describe the functionality of the CORDIET toolset. In section 5 we discuss the case studies and showcase the potential of our approach. In section 6 we elaborate on why a human-centered Knowledge Discovery in Databases (KDD) approach may be better suited for text mining than standard fully automated machine learning techniques.

2 CORDIET Project Setup

2.1 Student Groups

After several presentations of the text mining research we are doing and the software system we want to develop, over 20 bachelor and master students of Moscow Higher School of Economics (HSE) showed their interest to actively participate. After several plenary meetings where the overall specification of the system was discussed, each student chose a component he or she wanted to develop. The goal was not only to develop a working system but also to help students gain experience which is valuable for their future career. Each student was given the task to first collect relevant literature, existing open source implementations, etc. Then we discussed in detail specifications for the component, interoperability requirements with other student components, programming language and useful APIs, etc. Although for several components of our system open source implementations exist, we chose to let students re-implement them so they could fully master technical programming skills and the data analyses techniques they are working with.

2.2 Software Architecture

We chose for a three-layered client-server architecture. The majority of the computationally intensive tasks is performed by the server components which are made available through web services.

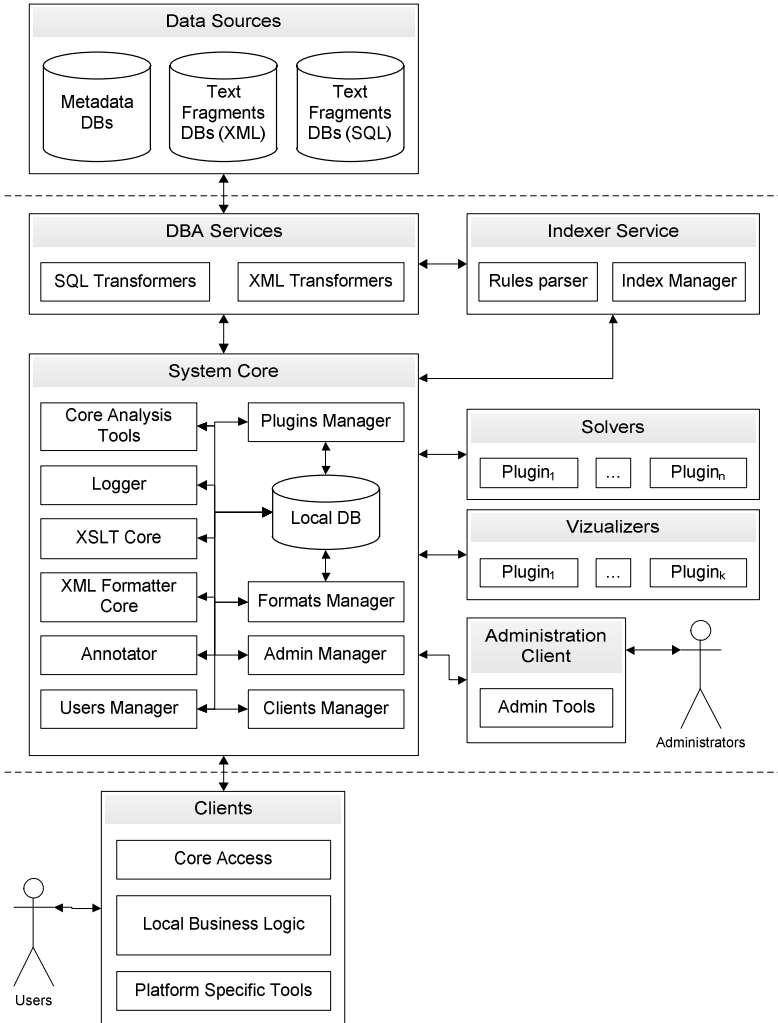


Fig. 1. A representation of the CORDIET architecture

The major components of the data layer are a relational database (at the moment we use PostgreSQL), XML input data files and the Lucene indexer. The data indexing component reads XML files from a selected dataset, parses these files into the SQL database and generates the Lucene index. The database content and Lucene index can be accessed and used by the business layer components through the data access layer. The

business layer will offer functionality to create business objects such as Hidden Markov Models, Emergent Self Organising Maps, concept lattices etc. (at the moment only the concept lattice module is fully implemented and working). The user can install the client module and remotely connect to the services which make functionality of the middle layer available. Figure 1 contains a high level overview of the software system.

3 Data Sources

In this research four main data sources have been used for empirical validation. The first data source was the police database “Basis Voorziening Handhaving” (BVH) of the Amsterdam-Amstelland police. Multiple datasets were extracted from this data source, including the domestic violence, human trafficking and terrorism dataset. The second data source was the World Wide Web, from which we collected 1072 scientific articles on Formal Concept Analysis (FCA). The third dataset consists of 148 breast cancer patients that were hospitalized in GZA hospital campus Sint-Augustinus during the period January 2008 till June 2008. The fourth dataset contains 533 chat conversations of pedophiles from the website www.perverted-justice.com.

Table 1. Overview of data sources used for empirically validating the CORDIET toolset

	textual	unstructured	data type	#data items	year
Domestic violence	X	X	incident reports	4814	2007
Human trafficking	X	X	observational reports	266157	2005-2010
Terrorism	X	X	observational reports	166577	2005-2009
Pedophiles	X	X	chat conversations	533	2004-2011
Clinical pathways	X		patient data	148	2008
Scientific articles	X	X	papers	1072	2003-2011

3.1 Data Source BVH

The database system BVH is used by all police forces of the Netherlands, the military police and the Royal Marechaussee. This database system contains both structured and unstructured textual information. The contents of the database are subdivided in two categories: incidents and activities. Incident reports describe events that took place which are in violation with the law. These include violence, environmental and financial crimes. During our first case study we analyzed 4814 incident reports describing violent incidents, filed in 2007, and we aimed at automatically recognizing the domestic violence cases.

Activities are often performed after certain incidents occurred and include interrogations, arrestments, etc., but activities can also be performed independent of any incident, such as motor vehicle inspections, an observation made by a police officer of a suspicious situation, etc. Each of these activities performed are described in a textual report by the responsible officer. In the year 2005, Intelligence Led Policing (Collier 2006) was introduced at the police of Amsterdam, resulting in a

sharp increase in the number of filed activity reports describing observations made by police officers, i.e. from 34817 in 2005 to 67584 in 2009. These observational reports contain a short textual description of what has been observed and may be of great importance for finding new criminals. In the second and third case study, we used the observations made by police officers to find indications for human trafficking in 266157 reports and indications for radicalizing behavior in 166577 reports. The involved persons and vehicles are stored in structured data fields in a separate database table and are linked to the unstructured report using relational tables. Therefore, we wrote an export program that automatically composes documents based on the most recently available information in the databases. These documents are stored in XML format and can be read by the CORDIET toolset.

3.2 Data Source Scientific Articles

Over 1000 pdf files containing articles about FCA research were downloaded from the WWW and analyzed with the CORDIET system. During the analysis, these pdf-files were converted to ordinary text and the abstract, title and keywords were extracted. Lucene was used to index the extracted parts of the papers using our thesaurus containing terms referring to interesting research topics. The result was a cross table describing the relationships between the papers and the research topics from the thesaurus. This cross table was used as a basis to generate the lattices.

We only used abstract, title and keywords because the full text of the paper may mention a number of concepts that are irrelevant to the paper. For example, if the author wrote an article on information retrieval but also gives an overview of related work mentioning papers on fuzzy FCA, rough FCA, etc., these concepts may be irrelevant however they are detected in the paper. If they are relevant to the entire paper we found they were typically also mentioned in title, abstract or keywords.

3.3 Data Source Clinical Pathways

The third dataset consists of 148 breast cancer patients that were hospitalized, in GZA hospital campus Sint-Augustinus, during the period from January 2008 till June 2008. They all followed the care trajectory determined by the clinical pathway Primary Operable Breast Cancer (POBC), which structures one of the most complex care processes in the hospital. Every activity or treatment step performed to a patient is logged in a database and in the dataset we included all the activities performed during the hospitalization of these patients. Each activity has a unique identifier and we have 469 identifiers in total for the clinical path POBC. We clustered activities with a similar semantic meaning to reduce the complexity of the lattices. The resulting dataset is a collection of XML files where each XML contains all activities performed to one patient.

3.4 Data Source Pedophiles

Because original chat data collected by the Dutch police force organizations is restricted by law, results may not be made public. To demonstrate our FCA based method we use

the chat data collected by a public American organization, Perverted Justice, which actively searches for pedophiles on the internet. We downloaded 533 chat files, i.e. one for each of the 533 different suspects. The victims in all chat files are adults playing the role of a young girl or boy in the age from 12 to 14. All these adults are members of the Perverted Justice organization and are trained to act as a youngster. The adults playing the victim try to lure the suspect by playing his or her role as good as possible. The behavior of the victims cannot be representative for young girls or boys, but the behavior of the suspects is realistic since they really believe to have contact with a young girl or boy and act in that way.

4 Functionality of the CORDIET Software

Figure 2 displays a screenshot of the CORDIET software which the user will see when he starts the system. First, the user can load a set of textual XML files (e.g. police reports) in the database (1). The structured part of these reports will be displayed together with their textual content on the right of the screen (2). One of the central components of our text analysis environment is the semantic network containing the collection of attributes used to index the data files. The initial semantic network is typically constructed based on expert prior knowledge and incrementally improved by analyzing the concept gaps and anomalies in the resulting lattices. The user can then edit the semantic network containing terms (e.g. my father, my mother, my sister), clusters of terms (e.g. family members), temporal (e.g. January till June 2009) and more complex compound attributes (e.g. “family members” and not “January till June 2009”) which will be used to analyze the texts with CORDIET. The semantic network contains multiple abstraction levels. The first level of granularity

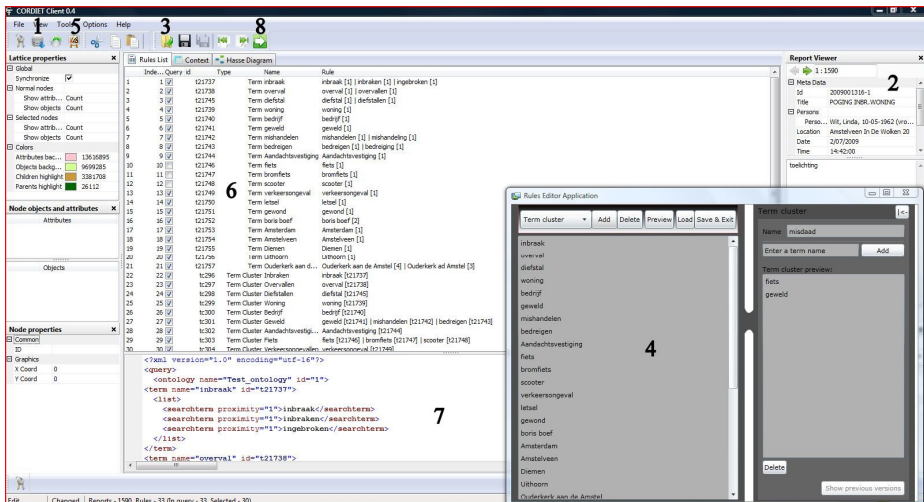


Fig. 2. CORDIET data preparation stage

contains the search terms of which most are grouped together based on their semantic meaning to form the term clusters at the second level of granularity. The compound attributes can be composed of simple attributes using first order logic operators. The user can open (3) a textual editor (4) or open (5) a separate graph based editor to work with this semantic network. Using the selection pane (6) he can select the attributes which will be used for analysis. The corresponding XML code used to store the ontology is shown below (7). By clicking on the green button (8) a formal context will be created from which other visual artifacts can be derived. In the current prototype only the concept lattice based algorithms have been implemented.

The user has several options to interact with the lattice (9) visualization in figure 3. He can choose to display the contents of objects and attributes but also to see their names in the lattice and use a condensed representation as in the lattice in Figure 3 (10). After clicking on a concept, the names of the texts (in this example police reports) in the extent of the concept are shown (12) together with the attribute names in the intent (11) on the left. After clicking on the name of a report, the contents will be visualized on the right (13). After clicking on the name of an attribute its components will also be visualized on the right (14). Clicking on a node will highlight all concepts on paths to the infimum and supremum of the lattice (see figure 5).

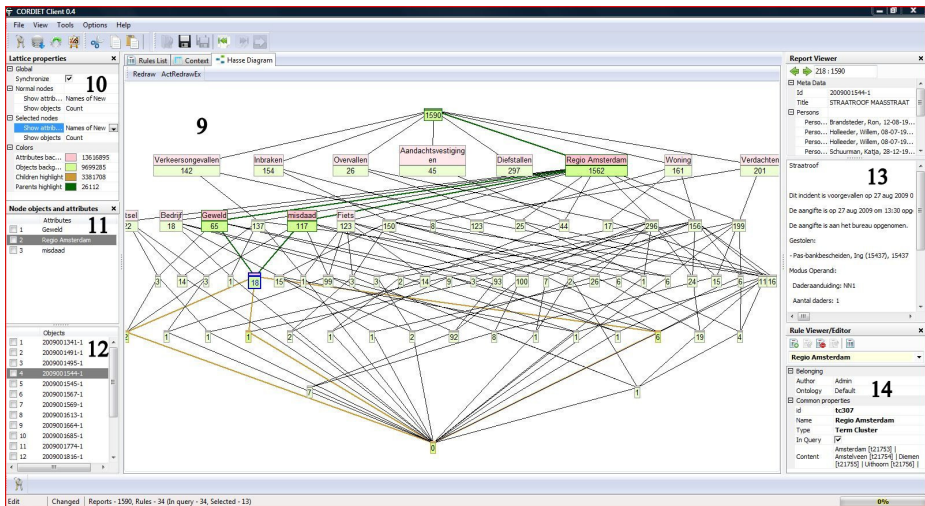


Fig. 3. CORDIET lattice interaction stage

5 Case Studies

In the healthcare case study (see section 5.1) in collaboration with GZA hospitals we worked on structured textual data, namely database logs of activities performed to patients to identify quality issues in care processes (Poelmans et al. 2010c). The other four case studies described in this section were a collaboration between KU Leuven and the Amsterdam-Amstelland police. In the domestic violence research (see section

5.2) we aimed at exploring and refining the concept and definition of domestic violence by analyzing statements made by victims to the police. An important spin-off of this exercise was the development of a highly accurate automated case labeling system (Poelmans et al. 2011a, Poelmans et al. 2010a). When we started analyzing observational reports the goal was to extract unknown suspects potentially involved in human trafficking, forced prostitution (see section 5.3 and 5.4) and terrorist activities (Poelmans et al. 2011b, Elzinga et al. 2010). We also investigated how we could offer police investigators quick but thorough insight in long chat conversations (see section 5.5) of potential pedophiles with children (Elzinga et al. 2012). Finally, we show how we used FCA as a meta-technique to analyze the literature on FCA (see section 5.6). In the remainder of this section we showcase briefly the potential of our software system for each of these cases. For a thorough description of these cases the reader is kindly referred to the papers mentioned.

5.1 Care Process Analysis

Care pathways are a methodology to structure multidisciplinary care processes of patients with a specific clinical problem. During auditing of the breast cancer care process in the GZA hospital group we obtained the concept lattice in Figure 4. The diagram shows that several mandatory key interventions were not always performed to hospitalized patients who underwent breast conserving surgery. For example, “physiotherapy”, “emotional support” and “counseling by social service” were not performed to 15, 2 and 3 of the 60 cancer patients respectively. After presenting these results to the care process managers, we jointly looked for the root causes of this

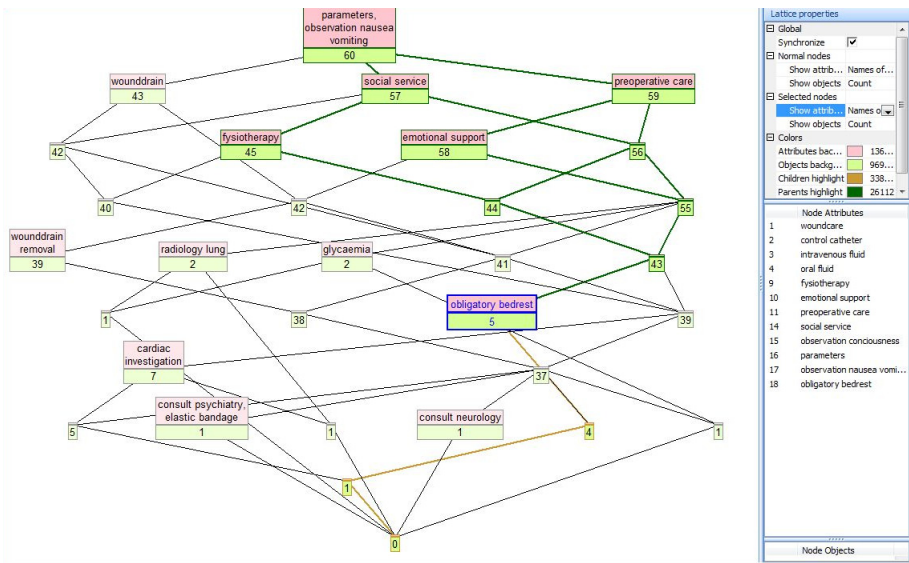


Fig. 4. Activities performed to breast cancer patients during hospitalization

problem. One of these causes turned out to be that over the past years the length of stay of the patients in the hospital was dramatically lowered without modifying the original care process model. After these findings the prescribed process model was rewritten to take into account this shorter length of stay.

5.2 Domestic Violence under Scrutiny

A definition of a problem is often inaccurate and incomplete due to the complex nature of the reality it was designed to deal with.

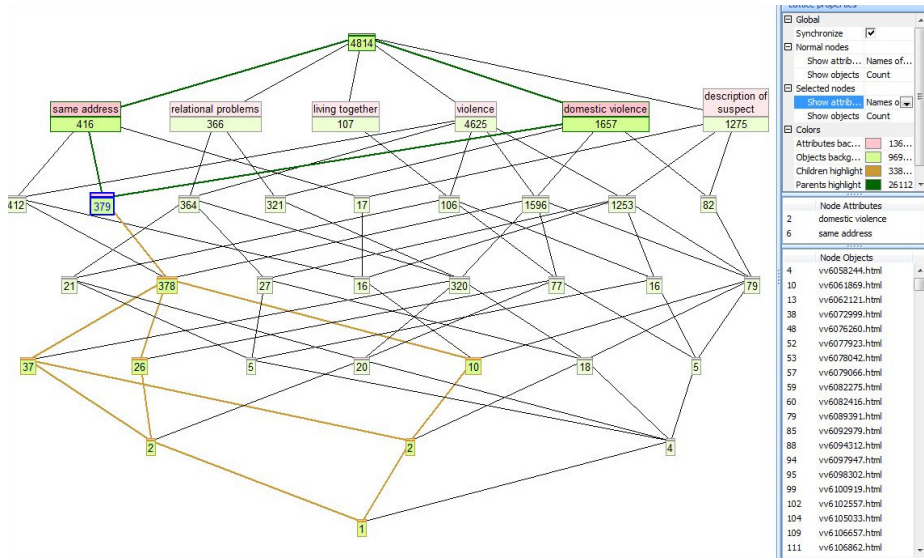


Fig. 5. Analyzing statements made by victims of a violent incident

An example is the domestic violence definition which was employed by the Amsterdam-Amstelland police (Keus et al. 2000): “*Domestic violence can be characterized as serious acts of violence committed by someone in the domestic sphere of the victim. Violence includes all forms of physical assault. The domestic sphere includes all partners, ex-partners, family members, relatives and family friends of the victim. The notion of family friend includes persons that have a friendly relationship with the victim and (regularly) meet with the victim in his/her home.*” The lattice in Figure 5 contains 4814 police reports of which 1657 were labeled as domestic violence by police officers.

With CORDIET, the user can visually represent the underlying concepts in the data, gain insight in the complexity of the domain under investigation and zoom in on interesting concepts. For example we clicked on the node with 379 reports where suspect and victim lived on the same address and labeled as domestic violence by officers. Domain experts assumed that a situation where perpetrator and victim live at the same address is always a case of domestic violence, since these persons are

probably family members, however this turned out not to be true. Analysis of the reports with attribute “same address” and not labeled as domestic violence revealed borderline cases such as violence in prisons, violence between a caretaker and inhabitant of an old folks home, etc. Each of these cases were presented to the steering board of the domestic violence policy. This resulted in an improved definition of domestic violence and an improved handling of domestic violence cases.

5.3 Identifying Human Trafficking Suspects

In the past, relevant suspects sometimes remained undetected in the overload of observational reports. Since human trafficking indications for observed persons are spread over multiple reports which are typically filed by different officers, a visual picture which summarizes these data and makes it accessible for exploration is an important instrument for investigators. With CORDIET, we present this picture to the user in the form of a concept lattice. For example Figure 6 shows potential human trafficking suspects, i.e. men who force girls to work in prostitution, with Eastern European nationality (Poelmans et al. 2011).

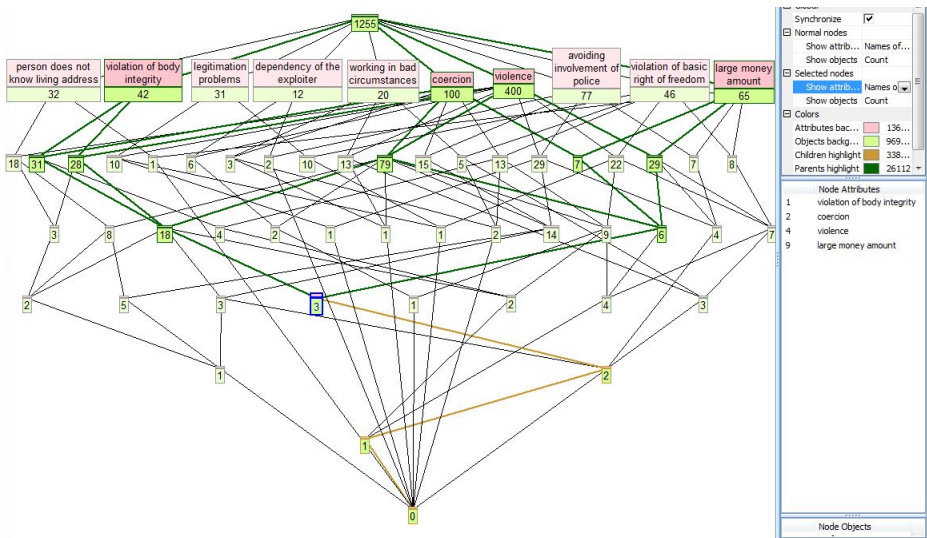


Fig. 6. Lattice of potential suspects and victims of human trafficking

We clicked on the concept with 3 persons in the extent and attributes “violation of body integrity”, “coercion”, “violence” and “large money amount” in the intent. The police officer can see the names of the persons in the “node objects” pane and double click on a name to create a detailed profile lattice for him or her (see also section 5.4). Please note that we made the node objects pane containing names of potential suspects and victims invisible. Similar results were achieved for terrorism data (Elzinga et al. 2010).

5.4 Profiling Human Trafficking Suspects

A profile of a selected potential suspect can be automatically generated using our system and displays all available information in a lattice together with the temporal evolution of this person (Poelmans et al. 2011b). Figure 7 shows a lattice profile of a loverboy suspect. The objects are names and birth dates of persons found in reports in which our main suspect was mentioned. After analyzing the lattice with domain experts it became clear that he used violence to make the 2 young girls (Sardientje and Hermina) consent to sexual exploitation.

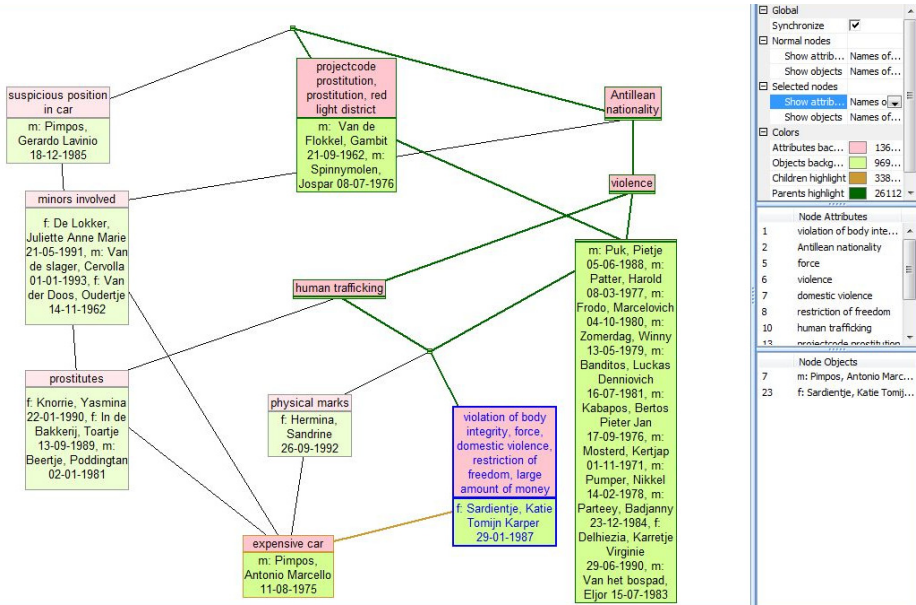


Fig. 7. Analysis of social network of suspect Pimpos Antonio Marcello (fictive name)

5.5 Analyzing Chat Conversations of Pedophiles

Chat conversations can be very long and time-consuming to read. A system which helps officers quickly identify those conversations posing a threat to a child’s safety and understand what has been talked about may significantly speed up and improve the efficiency of their work (Elzinga et al. 2012).

The lattice in Figure 8 shows how a set of 533 chat conversations was analyzed with FCA. We defined 7 term clusters containing keywords which were used by pedophiles in their chat conversations. We numbered these 7 attributes according to the severity of the threat to the child’s safety. We clicked on a concept with 96 conversations in the extent and attributes “asks”, “asks about sex”, “describes about sex” and “asks for address”. In the “node objects” pane the user can click on the name of a conversation to display its contents. In Elzinga et al. (2012) we describe in detail

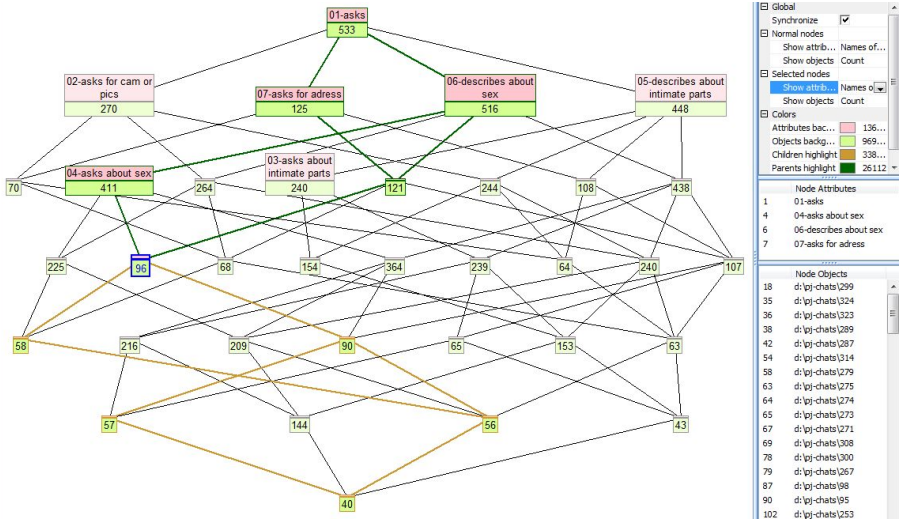


Fig. 8. Analyzing chat conversations of pedophiles with members of the perverted justice organization who pretend to be a young child

how we selected chats from such a concept lattice and analyze them in detail with temporal relational semantic systems.

5.6 FCA Literature Study

CORDIET was also used in an exploratory study for visually representing and exploring scientific papers (Poelmans et al. 2010b, Poelmans et al. 2012). The user can dynamically select and deselect attributes representing research topics and relevant papers will be shown. An author looking for relevant works in his/her filed may benefit from such a system.

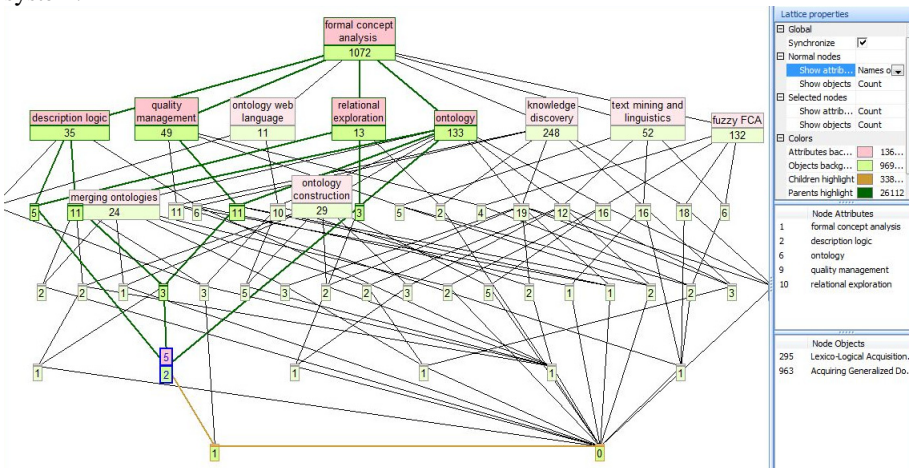


Fig. 9. Analyzing 1072 papers on Formal Concept Analysis with a zoom on ontology engineering papers

The lattice in Figure 9 displays 1072 papers on Formal Concept Analysis. The attributes which we chose to analyze these data are related to ontology engineering. The selected node contains 2 papers on quality management of ontologies using relational exploration, an algorithm which can be applied in the context of Formal Concept Analysis enriched with description logics.

6 Discussion, Conclusions and Future Work

Yearly more than 5000 statements are made by victims of a violent incident to the Amsterdam-Amstelland police and over 60000 observational reports are filed by officers. The need for a text analysis system became apparent since there is no capacity to deal with all this information manually. Existing fully automated text mining methods were found to be no good option. First, automated machine learning techniques such as decision trees, support vector machines, neural networks, automated keyword extraction techniques, etc. assume that the underlying concepts of the domain are clear, can be extracted from the data and used for classification. Unfortunately we found this is often not true, e.g. there was no consensus amongst police officers on whether certain borderline cases should be labeled as domestic violence or non-domestic violence, officers labeled several cases wrongly, prior knowledge was not always useful, etc. Second, in observational police reports, indications against a certain suspect are spread over multiple reports. In the case of human trafficking for example, an additional complicating factor is that only a few thousand reports in a dataset of over 250 000 reports are relevant. Third, often a human expert should stay in the loop because of the particular nature of the domain under investigation. Selecting suspects for in-depth investigation should be done by a human expert who can be held accountable for his decisions.

In each of the described case studies FCA was used to create intuitive and interactive visualizations which can easily be interpreted by domain experts. To cope with scalability issues, attributes can be clustered, segments of the data can be selected, objects can be grouped, etc. in the CORDIET system. In particular the data summarization capabilities of the lattice diagrams were found to be of interest to the users. Texts, their properties, connections with other pieces of text, etc. can be distilled from such diagrams with ease. We believe that the presented CORDIET system may significantly improve the efficiency of working with unstructured textual data. However, we are aware that a lot of work remains to be done. Avenues for future research include:

- Implementation of other (complementary) data visualization techniques such as Emergent SOM and HMMs and integrating these models with the existing concept lattice module and with each other such that data exploration can be done in an efficient yet thorough manner.
- Analyzing the possibilities of NLP techniques for Dutch language to speed up thesaurus building.
- Extending existing functionality to better analyze social structure of criminal communities.

Acknowledgments. Jonas Poelmans is Aspirant of the “Fonds voor Wetenschappelijk Onderzoek – Vlaanderen” (FWO) or Research Foundation – Flanders.

References

- [1] Cimiano, P., Hotho, A., Staab, S.: Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *J. Artif. Intell. Res (JAIR)* 24, 305–339 (2005)
- [2] Collier, P.M.: Policing and the intelligent application of knowledge. *Public Money & Management* 26(2), 109–116 (2006)
- [3] Elzinga, P., Poelmans, J., Viaene, S., Dedene, G., Morsing, S.: Terrorist threat assessment with Formal Concept Analysis. In: *Proc. IEEE International Conference on Intelligence and Security Informatics*, Vancouver, Canada, May 23–26, pp. 77–82 (2010)
- [4] Elzinga, P., Wolff, K.E., Poelmans, J., Viaene, S., Dedene, G.: Analyzing chat conversations of arrested child abusers with temporal relational semantic systems. In: *Contributions to 10th International Conference on Formal Concept Analysis*, Leuven, Belgium, May 6–10 (2012)
- [5] Keus, R., Kruijff, M.S.: *Huiselijk geweld, draaiboek voor de aanpak*. Directie Preventie, Jeugd en Sanctiebeleid van de Nederlandse justitie (2000)
- [6] Maio, C.D., Fenza, G., Gaeta, M., Loia, V., Orciuoli, F., Senatore, S.: RSS-based e-learning recommendations exploiting fuzzy FCA for Knowledge Modeling. *Applied Soft Computing* 12(1), 113–124 (2012)
- [7] Poelmans, J., Dedene, G., Verheyden, G., Van der Mussele, H., Viaene, S., Peters, E.: Combining Business Process and Data Discovery Techniques for Analyzing and Improving Integrated Care Pathways. In: Perner, P. (ed.) *ICDM 2010*. LNCS, vol. 6171, pp. 505–517. Springer, Heidelberg (2010c)
- [8] Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: A Case of Using Formal Concept Analysis in Combination with Emergent Self Organizing Maps for Detecting Domestic Violence. In: Perner, P. (ed.) *ICDM 2009*. LNCS, vol. 5633, pp. 247–260. Springer, Heidelberg (2009)
- [9] Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: Curbing domestic violence: Instantiating C-K theory with Formal Concept Analysis and Emergent Self Organizing Maps. *Intelligent Systems in Accounting, Finance and Management* 17(3–4), 167–191 (2010a)
- [10] Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: Formal Concept Analysis in Knowledge Discovery: A Survey. In: Croitoru, M., Ferré, S., Lukose, D. (eds.) *ICCS 2010*. LNCS, vol. 6208, pp. 139–153. Springer, Heidelberg (2010b)
- [11] Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: Formally Analyzing the Concepts of Domestic Violence. *Expert Systems with Applications* 38(4), 3116–3130 (2011a)
- [12] Poelmans, J., Elzinga, P., Dedene, G., Viaene, S., Kuznetsov, S.O.: A Concept Discovery Approach for Fighting Human Trafficking and Forced Prostitution. In: Andrews, S., Polovina, S., Hill, R., Akhgar, B. (eds.) *ICCS-ConceptStruct 2011*. LNCS, vol. 6828, pp. 201–214. Springer, Heidelberg (2011b)
- [13] Poelmans, J., Ignatov, D.I., Viaene, S., Dedene, G., Kuznetsov, S.: Text mining scientific papers: a survey on FCA-based information retrieval research. In: *12th Industrial Conference on Data Mining*. LNCS, July 13–20, Berlin, Germany. Springer (2012)
- [14] Rabiner, L.R.: A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings IEEE* 77(2), 257–286 (1989)

- [15] Stumme, G., Wille, R., Wille, U.: Conceptual Knowledge Discovery in Databases using Formal Concept Analysis Methods. In: PKDD 1998. LNCS, vol. 1510, pp. 450–458. Springer, Heidelberg (1998)
- [16] Ultsch, A.: Maps for visualization of high-dimensional Data Spaces. In: Proc. WSOM 2003, Kyushu, Japan, pp. 225–230 (2003)
- [17] Ultsch, A., Hermann, L.: Architecture of emergent self-organizing maps to reduce projection errors. In: Proc. ESANN 2005, pp. 1–6 (2005)
- [18] Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) *Ordered Sets*, pp. 445–470. Reidel, Dordrecht-Boston (1982)
- [19] Wolff, K.E.: States, Transitions, and Life Tracks in Temporal Concept Analysis. In: Ganter, B., Stumme, G., Wille, R. (eds.) *Formal Concept Analysis. LNCS (LNAI)*, vol. 3626, pp. 127–148. Springer, Heidelberg (2005)

Text Mining Scientific Papers: A Survey on FCA-Based Information Retrieval Research

Jonas Poelmans^{1,4}, Dmitry I. Ignatov⁴, Stijn Viaene^{1,2},
Guido Dedene^{1,3}, and Sergei O. Kuznetsov⁴

¹ K.U. Leuven, Faculty of Business and Economics, Naamsestraat 69,
3000 Leuven, Belgium

² Vlerick Leuven Gent Management School, Vlamingenstraat 83,
3000 Leuven, Belgium

³ Universiteit van Amsterdam Business School, Roetersstraat 11
1018 WB Amsterdam, The Netherlands

⁴ National Research University Higher School of Economics (HSE), Pokrovsky boulevard 11,
101000 Moscow, Russia

{Jonas.Poelmans, Stijn.Viaene, Guido.Dedene}@econ.kuleuven.be,
{dignatov, skuznetsov}@hse.ru

Abstract. Formal Concept Analysis (FCA) is an unsupervised clustering technique and many scientific papers are devoted to applying FCA in Information Retrieval (IR) research. We collected 103 papers published between 2003-2009 which mention FCA and information retrieval in the abstract, title or keywords. Using a prototype of our FCA-based toolset CORDIET, we converted the pdf-files containing the papers to plain text, indexed them with Lucene using a thesaurus containing terms related to FCA research and then created the concept lattice shown in this paper. We visualized, analyzed and explored the literature with concept lattices and discovered multiple interesting research streams in IR of which we give an extensive overview. The core contributions of this paper are the innovative application of FCA to the text mining of scientific papers and the survey of the FCA-based IR research.

1 Introduction

According to Manning et al. (2008), “information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).” In the past, only specialized professions such as librarians had to retrieve information on a regular basis. These days, massive amounts of information are available on the www and hundreds of millions of people make use of information retrieval systems such as web or email search engines on a daily basis. Formal Concept Analysis (FCA) was introduced in the early 1980s by Rudolf Wille as a mathematical theory (Wille 1982) and is a popular technique within the IR field. FCA is concerned with the formalization of concepts and conceptual thinking and has been applied in many disciplines such as software engineering, knowledge discovery and ontology construction during the last 15 years. The core contributions of this paper are as follows. We visually represent

the literature on FCA and IR using concept lattices, in which the objects are the scientific papers and the attributes are the relevant terms available in the title, keywords and abstract of the papers. We developed a toolset with a central FCA component that we use to index the papers with a thesaurus containing terms related to FCA research and to generate the lattices. We zoom in and give an extensive overview of the papers published between 2003 and 2009 on using FCA in information retrieval.

The remainder of this paper is composed as follows. In section 2 we introduce the essentials of FCA theory and the knowledge browsing environment we developed to support this literature analysis. In section 3 we describe the dataset used. In section 4 we visualize the FCA literature on information retrieval using FCA lattices and we summarize the papers published in this field. Section 5 concludes the paper.

2 Formal Concept Analysis

FCA (Ganter et al. 1999, Wille 1982) is a well established technique in mathematics and computer science and multiple partial surveys were published during the past years. A textual overview of part of the literature published until the year 2004 on FCA is given by Priss (2006). An overview of available FCA software is provided by Tilley (2004) and in Tilley et al. (2007), an overview of 47 FCA-based software engineering papers is given. The authors categorized these papers according to the 10 categories as defined in the ISO 12207 software engineering standard and visualized them in a concept lattice. In Lakhali et al. (2005), a survey on FCA-based association rule mining techniques is given. Poelmans et al. (2010) give an extensive overview of KDD applications of FCA. FCA groups scientific papers containing terms from the same term-clusters in concepts. The starting point of the analysis is a formal context (G, M, I) consisting of rows G (i.e. objects), columns M (i.e. attributes) and crosses $I \subseteq G \times M$ (i.e. relationships between objects and attributes).

Table 1. Example of a formal context

	browsing	mining	Software	web services	FCA	information retrieval
Paper 1	X	X	X		X	
Paper 2			X		X	X
Paper 3		X		X	X	
Paper 4	X		X		X	
Paper 5				X	X	X

An example of a cross table is displayed in Table 1. In the latter, scientific papers (i.e. the objects) are related (i.e. the crosses) to a number of terms (i.e. the attributes); here a paper is related to a term if the title or abstract of the paper contains this term. Given a formal context, FCA then derives all concepts from this context and orders them according to a subconcept-superconcept relation, resulting in a lattice.

The notion of concept is central to FCA. The way FCA looks at concepts is in line with the international standard ISO 704, that formulates the following definition: “A concept is considered to be a unit of thought constituted of two parts: its extent and its

intent.” The extent consists of all objects belonging to the concept, while the intent comprises all attributes shared by those objects. Let us illustrate the notion of concept of a formal context using the data in Table 1. For a set of objects $O \subseteq G$, the common features can be identified, written O' , via:

$$A = O' = \{m \in M \mid \forall o \in O : (o, m) \in I\}$$

Take the attributes that describe paper 4 in Table 1, for example. By collecting all papers of this context that share these attributes, we get to a set $O \subseteq G$ consisting of papers 1 and 4. This set O of objects is closely connected to set A consisting of the attributes “browsing”, “software” and “FCA”: $O = A' = \{o \in G \mid \forall a \in A : (o, a) \in I\}$

That is, O is the set of all objects sharing all attributes of A , and A is the set of all attributes that are valid descriptions for all the objects contained in O . Each such pair (O, A) is called a formal concept (or concept) of the given context. The set $A = O'$ is called the intent, while $O = A'$ is called the extent of the concept (O, A) .

There is a natural hierarchical ordering relation between the concepts of a given context that is called the subconcept-superconcept relation.

$$(O_1, A_1) \leq (O_2, A_2) \Leftrightarrow (O_1 \subseteq O_2 \Leftrightarrow A_2 \subseteq A_1)$$

A concept $d = (O_1, A_1)$ is called a subconcept of a concept $e = (O_2, A_2)$ (or equivalently, e is called a superconcept of a concept d) if the extent of d is a subset of the extent of e (or equivalently, if the intent of d is a superset of the intent of e). For example, the concept with intent “browsing”, “software”, “mining” and “FCA” is a subconcept of a concept with intent “browsing”, “software” and “FCA.” With reference to Table 1, the extent of the latter is composed of papers 1 and 4, while the extent of the former is composed of paper 1.

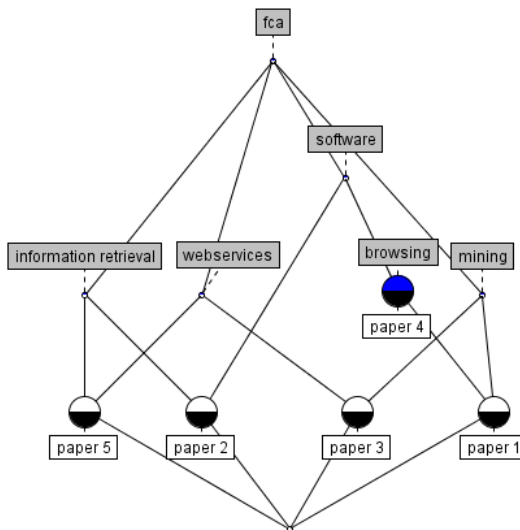


Fig. 1. Line diagram corresponding to the context from Table 1

The set of all concepts of a formal context combined with the subconcept-superconcept relation defined for these concepts gives rise to the mathematical structure of a complete lattice, called the concept lattice of the context. The line diagram in Figure 1 is a representation of the concept lattice of the formal context abstracted from Table 1. The circles or nodes in this line diagram represent the formal concepts. The shaded boxes (upward) linked to a node represent the attributes used to name the concept. The non-shaded boxes (downward) linked to the node represent the objects used to name the concept. The information contained in the formal context of Table 1 can be distilled from the line diagram in Figure 1 by applying the following reading rule: An object “g” is described by an attribute “m” if and only if there is an ascending path from the node named by “g” to the node named by “m.” For example, paper 1 is described by the attributes “browsing”, “software”, “mining” and “FCA.”

We developed a knowledge browsing environment CORDIET to support our literature analysis process (Poelmans et al. 2010b, Poelmans et al. 2010c). One of the central components of our text analysis environment is the thesaurus containing the collection of terms describing the different research topics. The initial thesaurus was constructed based on expert prior knowledge and was incrementally improved by analyzing the concept gaps and anomalies in the resulting lattices. The thesaurus is a layered thesaurus containing multiple abstraction levels. The first and finest level of granularity contains the search terms of which most are grouped together based on their semantical meaning to form the term clusters at the second level of granularity.

The papers that were downloaded from the World Wide Web (WWW) were all formatted in pdf. These pdf-files were converted to ordinary text and the abstract, title and keywords were extracted. The open source tool Lucene was used to index the extracted parts of the papers using the thesaurus. The result was a cross table describing the relationships between the papers and the term clusters or research topics from the thesaurus. This cross table was used as a basis to generate the lattices.

3 Dataset

This Systematic Literature Review (SLR) has been carried out by considering a total of 103 papers related to FCA and IR published between 2003 and 2009 in the literature and extracted from the most relevant scientific sources. The sources that were used in the search for primary studies contain the work published in those journals, conferences and workshops which are of recognized quality within the research community. These sources are: *IEEE Computer Society*, *ACM Digital Library*, *Sciencedirect*, *Springerlink*, *EBSCOhost*, *Google Scholar*, *Conference repositories: ICFCA, ICCS and CLA conference*. Other important sources such as DBLP or CiteSeer were not explicitly included since they were indexed by some of the mentioned sources (e.g. Google Scholar). In the selected sources we used various search strings including “Formal Concept Analysis”, “FCA”, “concept lattices”, “Information Retrieval”. To identify the major categories for the literature survey we also took into account the number of citations of the FCA papers at CiteseerX.

4 FCA-Based Information Retrieval Research

In Conceptual Knowledge Processing (CKP) the focus lies on developing methods for processing information and knowledge which stimulate conscious reflection, discursive argumentation and human communication (Wille 2006, Eklund et al. 2007). The word “conceptual” underlines the constitutive role of the thinking, arguing and communicating human being and the term “processing” refers to the process in which something is gained which may be knowledge. FCA can be particularly suited for IR because of its human-centeredness. The efficient retrieval of relevant information is promoted by the FCA representation that makes the inherent logical structure of the information transparent. FCA can be used for multiple purposes in IR (Priss 2006). First, FCA is an interesting instrument for browsing through large document collections. FCA can also support query refinement. Because a document-term lattice structures the available information as clusters of related documents which are partially ordered, lattices can be used to make suggestions for query enlargement in cases where too few documents are retrieved and for query refinement in cases where too many documents are retrieved. Third, lattices can be used for querying and navigation. An initial query corresponds to a start node in a document-term lattice. Users can then navigate to related nodes. Further, queries are used to “prune” a document-term lattice to help users focus their search (Carpineto et al. 1996b). For many purposes, some extra facilities are needed such as processing large document collections quickly, allowing more flexible matching operations, allowing ranked retrieval and give contextual answers to user queries. The past years many FCA researchers have also devoted attention to these issues.

The first attempts to use lattices for information retrieval are summarized in Priss (2000), but none of them resulted in practical implementations. Godin et al. (1989) developed a textual information retrieval system based on document-term lattices but without graphical representations of the lattices. The authors also compared the system's performance to that of Boolean queries and found that it was similar to and even better than hierarchical classification (Godin et al. 1993). They also worked on software component retrieval (Mili et al. 1997). In Carpineto et al. (2004a), their extensive work on information retrieval was summarized.

86 % of the papers on FCA and information retrieval are covered by the research topics in Figure 2. In section 4.1 and 4.2 we intuitively introduce the process of transforming data repositories into browsable FCA representations and performing query expansion and refinement operations. In section 4.3 and 4.4, the 28 % of papers on using FCA for representation of and navigation in image, service, web, etc. document collections are described. Defining and processing complex queries covers 6% of the papers and is described in section 4.5. Section 4.6 summarizes the papers on contextual answers (6% of papers) and ranking of query results (6% of papers).

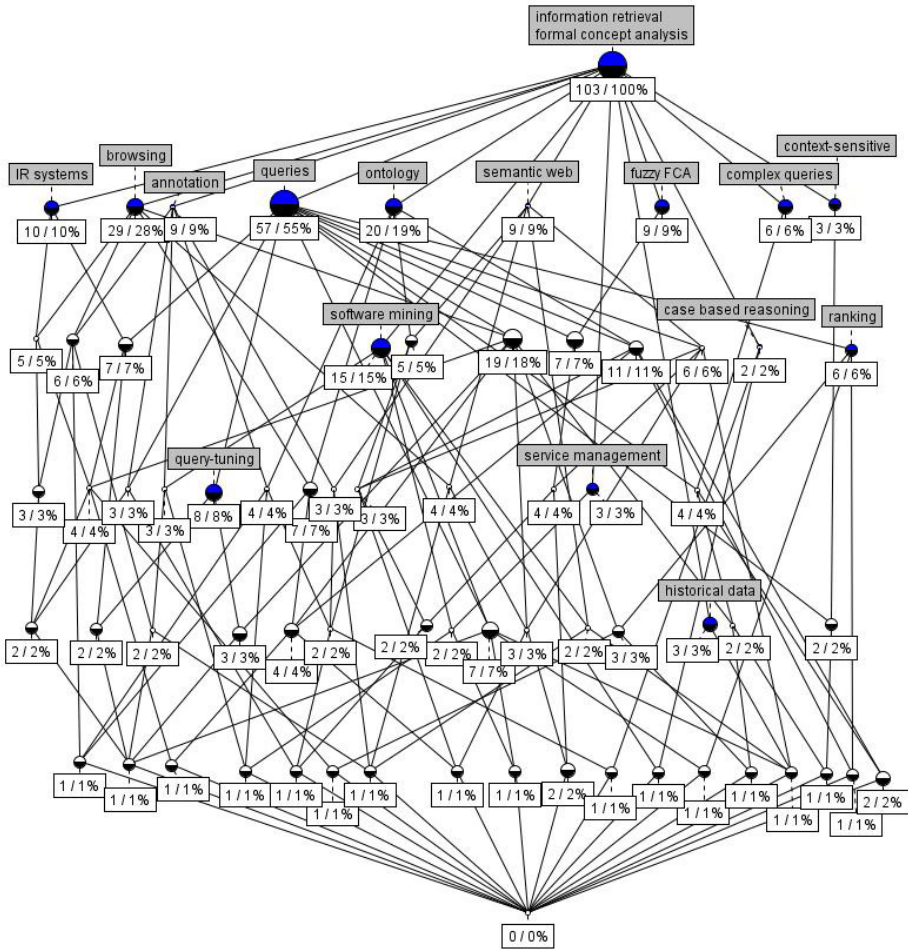


Fig. 2. Lattice containing 103 papers on using FCA in IR

4.1 Knowledge Representation and Browsing with FCA

In 28 % of the 103 selected papers, FCA is used for browsing and navigation through document collections. In more than half of these papers (18% of total number of papers), a combination of navigation and querying based on the FCA lattices is proposed. Annotation of documents and finding optimal document descriptors play an important role in effective information retrieval (9% of papers). All FCA-based approaches for information retrieval and browsing through large data repositories are based on the same underlying model. We first have the set G containing objects such as web pages, web services, images or other digitally available items. The set A of attributes can consist of terms, tags, descriptions, etc. These attributes can be related to certain objects through a relation $I \subseteq G \times M$ which indicates the terms, tags, etc.

can be used to describe the data elements in G . This triple (G, M, I) is a formal context from which the concept lattice can be created. The mathematical details of such a concept lattice are described in section 2.1. The process of obtaining a browsable FCA representation from such data is displayed in Fig. 3.

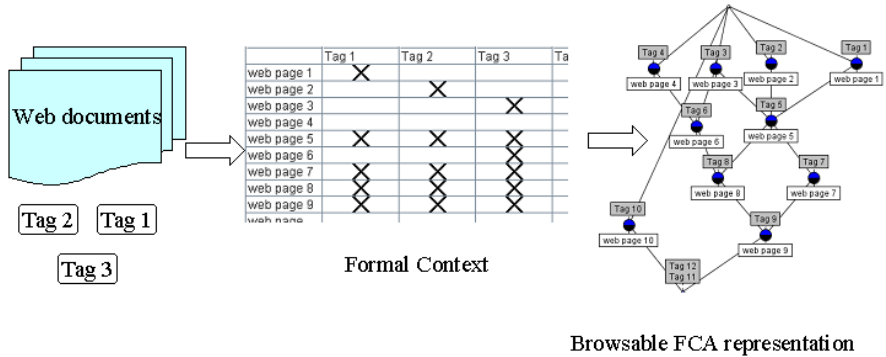


Fig. 3. Data transformation process

4.2 Query Result Improvement with FCA

Search engines are increasingly being used by amongst others web users who have an information need. The intent of a concept in an FCA lattice corresponds to a query and the extent contains the search results. A query ΛA uses a set of terms A and the system returns the answer by evaluating A' . Upon evaluating a query ΛA the system places itself on the concept (A', A'') which becomes the current concept c . For example in Fig. 4, the intent of the current concept $A_c = \{\text{Tag 1, Tag 2, Tag 3, Tag 4, Tag 5, Tag 6, Tag 8}\}$ and the extent of the current concept $O_c = \{\text{web page 8, web page 9}\}$. Since a query provided by a user only approximates a user's need, many techniques have been developed to expand and refine query terms and search results. Query tuning is the process of searching for the query that best approximates the information need of the user. Query refinements can help the user express his original need more clearly. Query refinement can be done by going to a lower neighbor of the current concept in the lattice by adding a new term to the query items. The user can navigate for example to a subconcept $((A_c \cup \{t\})', (A_c \cup \{t\})'')$ by adding term t .

Query enlargement, i.e. retrieving additional relevant web pages, can be performed by navigating to an upper neighbor of the current concept in the lattice by removing a term from the query items. The user can navigate for example to a superconcept $((O_c \cup \{o\})'', (O_c \cup \{o\})')$ by adding object o . The combination of subsequent refine and expand operations can be seen as navigation through the query space. Typically, navigation and querying are two completely separate processes, and the combination of both results in a more flexible and user-friendly method. These topics are investigated in 8 % of the IR papers.

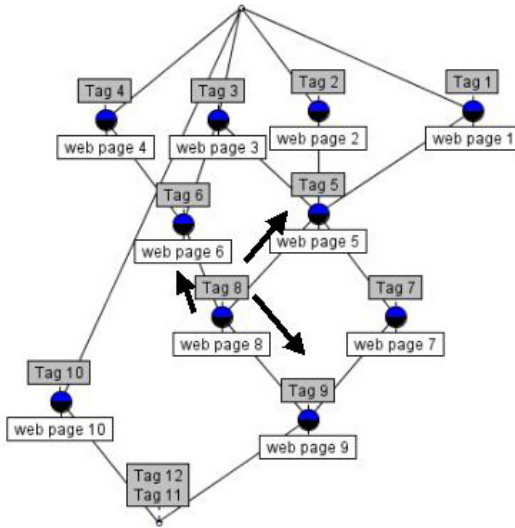


Fig. 4. Query tuning: from the current concept: an upward arrow for query expansion and a downward arrow for query refinement

4.3 Web and Email Retrieval

FCA has been used as the basis for many web-based knowledge browsing systems developed during the past years. Especially its comprehensible visualization capabilities seem to be of interest to the authors of these papers.

The results returned by web search engines for a given query are typically formatted as a list of URLs accompanied by a document title and a short summary of the document. Several FCA based systems were developed for analyzing and exploring these search results. CREDO (Carpineto et al. 2004), FooCA (Koester 2005, Koester 2006) and SearchSleuth (Ducrou et al. 2007, Dau et al. 2008) build a context for each individual query which contains the result of the query as objects and the terms found in the title and summary of each result as attributes. The CREDO system then builds an iceberg lattice which is represented as a tree and can be interactively explored by the user. FooCA shows the entire formal context to the user and offers a great degree of flexibility in exploring this table using the ranking of attributes, selecting the number of objects and attributes, applying stemming and stop word removal etc. SearchSleuth does not display the entire lattice but focuses on the search concept, i.e. the concept derived from the query terms. The user can easily navigate to its upper and lower neighbors and siblings. Nauer et al. (2009) also propose to use FCA for iteratively and interactively analyzing web search results. The user can indicate which concepts are relevant and which ones are not for the retrieval task. Based on this information the concept lattice is dynamically modified. Their research resulted in the CreChainDo system. Kim et al. (2004) presented the FCA-based document navigation system KAnavigator for small web communities in specialized domains. Relevant documents can be annotated with keywords by the users. Kim et al. (2006) extended

the search functionality by combining lattice-based browsing with conceptual scales to reduce the complexity of the visualization. Cigarran et al. (2004) present the JBraindead IR System which combines free-text search with FCA to organize the results of a query.

Cole et al. (2003) discuss a document discovery tool named Conceptual Email Manager (CEM) which is based on FCA. The program allows users to navigate through emails using a visual lattice. The paper also discusses how conceptual ontologies can support traditional document retrieval systems and aid knowledge discovery in document collections. The development of this software is based on earlier research on retrieval of information from semi-structured texts (Cole et al. 2001, Cole et al. 2000). Building further on this work is the Mail-Sleuth software (Eklund et al. 2004) which can be used to mine large email archives. Eklund et al. (2005) use FCA for displaying, searching and navigating through help content in a help system.

Stojanovic (2005) present an FCA-based method for query refinement that provides a user with the queries that are “nearby” the given query. Their approach for query space navigation was validated in the context of searching medical abstracts. Stojanovic (2004) presents the SMART system for navigation through an on-line product catalog. The products in the database are described by elements of an ontology and visualized with a lattice, in which users can navigate from a very general product-attribute cluster containing a lot of products to very specific clusters that seem to contain a few, but for the user highly relevant products. Spyrtos et al. (2006) describe an approach for query tuning that integrates navigation and querying into a single process. The FCA lattice serves for navigation and the attributes for query formulation. Le Grand et al. (2006) present an IR method based on FCA in conjunction with semantics to provide contextual answers to web queries. An overall lattice is built from tourism web pages. Then, users formulate their query and the best-matching concepts are returned, users may then navigate within the lattice by generalizing or refining their query. Eklund et al. (2008) present AnnotationSleuth to extend a standard search and browsing interface to feature a conceptual neighborhood centered on a formal concept derived from curatorial tags in a museum management system.

Cigarran et al. (2005) focus on the automatic selection of noun phrases as documents descriptors to build an FCA based IR system. Automatic attribute selection is important when using FCA in a free text document retrieval framework. Optimal attributes as document descriptors should produce smaller, clearer and more browsable concept lattices with better clustering features. Garcia et al. (2006) use FCA to perform semantic annotation of web pages with domain ontologies. Similarity matching techniques from Case Based Reasoning can be applied to retrieve these annotated pages as cases. Liu et al. (2007) use FCA to optimize a personal news search engine to help users obtain the news content they need rapidly. The proposed technique combines the construction of user background using FCA, the optimization of query keywords based on the user's background and a new layout strategy of search results based on a “Concept Tree”. Lungley et al. (2009) use implicit user feedback for adapting the underlying domain model of an intranet search system. FCA is used as an interactive interface to identify query refinement terms which help achieve better document descriptions and more browsable lattices.

4.4 Image, Software and Knowledge Base Retrieval

Another domain in which FCA has been applied as an information retrieval technique is software engineering. Efficient service management including the classification, semantic annotation and retrieval of web services are important challenges in service-centric software engineering. Peng et al. (2005) present a method for generating a concept lattice depicting conceptual relationships between web services and to accurately retrieve web services from these lattices. Bruno et al. (2005) propose an approach based on FCA and Support Vector Machines to automatically identify key concepts inside service textual documentation, build a lattice based on these service annotations and classify services to specific domains. Poshyvanyk et al. (2007) use a combination of FCA and Latent Semantic Indexing (LSI) for concept location in the source code of Eclipse. LSI is used to map the concepts expressed in queries to relevant parts of the source code. The result is a ranked list of source code elements, organized in an FCA lattice. Muangon et al. (2009) combine FCA with Case Based Reasoning (CBR) for choosing appropriate design patterns for a specific design problem. This approach solves some of the problems of existing design pattern search methods using keyword-search. Design patterns are applied to solve recurring software design problems. Peng et al. (2007) propose a method for the incremental construction of a component retrieval ontology based on FCA. The ontology contains the characterizations of the components stored in the repository.

Ahmad et al. (2003) build concept lattices from descriptions associated to images for searching and retrieving relevant images from a database. In the ImageSleuth project (Ducrou et al. 2006), FCA was also used for clustering of and navigation through annotated collections of images. The lattice diagram is not directly shown to the user. Only the extent of the present concept containing thumbnails, the intent containing image descriptions and a list of upper and lower neighbors is shown. In Ducrou (2007), the author built an information space from the Amazon.com online store and used FCA to discover conceptually similar DVDs and explore their conceptual neighborhood. The system was called DVDSleuth. Amato et al. (2008) start from an initial image given by the user and use a concept lattice for retrieving similar images. The attributes in this lattice are facets, i.e. an image similarity criterion based on e.g. texture, color or shape. The values in the context indicate for each facet how similar an image in the database is with respect to the user provided initial image. By querying, the user can jump to any cluster of the lattice by specifying the criteria that the sought cluster must satisfy. By navigation from any cluster, the user can move to a neighbor cluster, thus exploiting the ordering amongst clusters.

Ducrou et al. (2005b) presented an FCA-based application, D-SIFT, for exploring relational database schema. Tane et al. (2005) introduced the query-based multi context theory, which allows defining a virtual space of FCA-based views on ontological data. Tane et al. (2006) discuss the benefits of the browsing framework for knowledge bases based on supporting the user in defining pertinent views. Hachani et al. (2009) use fuzzy FCA to explain the reasons of a failed database query and generate the nearest subqueries with non-empty answers.

4.5 Defining and Processing Complex Queries with FCA

Multiple techniques have been developed to define and process complex queries and to integrate data coming from heterogeneous sources. This topic is discussed in 6% of the IR papers. De Souza et al. (2004) use FCA for processing user queries over a set of overlapping ontologies, which have been created by independent groups adopting different configurations for ontology concepts. For example in bioinformatics, it is often difficult to relate the resources with a user query since the query needs to be processed and distributed over several heterogeneous data sources. Messai et al. (2005) present an approach based on FCA to search relevant bioinformatics data sources for a given user query. Nafkha et al. (2005b) investigate the possibilities of using FCA for searching similar objects in heterogeneous information sources. Pollalilon et al. (2007) present a method based on FCA to provide contextual answers to user's queries from and to help their navigation in heterogeneous data sources. Cerauolo et al. (2007) use FCA for matching and mapping elements from heterogeneous data sources to a common ontology. Hitzler et al. (2006) present a new query language which allows querying formal contexts by means of logic programs written over attributes and objects.

4.6 Domain Knowledge in Search Results: Contextual Answers and Ranking

In this section, we discuss some of the techniques devised to provide contextual answers to user's queries and to incorporate domain knowledge into the organization of search results. Amongst others Carpineto et al. (2005) state that the main advantage of FCA for IR is the possibility of eliciting context and giving contextual answers to user's queries. This topic is discussed in 9% of the IR papers. Please note that several papers in section 4.3 and 4.4 present methods which provide the user contextual answers, here we focus on ranking of query search results. Several researchers use FCA lattices for measuring query-document relevance, i.e. concept lattice-based ranking (CLR). Messai et al. (2008) partially order the set of attributes with respect to their importance. This hierarchy represents domain knowledge used to improve lattice-based querying and navigation. Hierarchies of attributes are used to define complex queries containing attributes with different levels of importance. Zhang et al. (2008) propose a method based on FCA to build a two-level hierarchy for retrieved search results of a query to facilitate browsing the collection. After formal concepts are extracted using FCA, the concepts most relevant to the query are further extracted. Finally, Ignatov et al. (2009) use FCA for near-duplicate detection in web search results.

5 Conclusions

Since its introduction in 1982 as a mathematical technique, FCA became a well-known instrument in computer science. Over 700 papers have been published over the past 7 years on FCA and 103 of them showed the method's usefulness for IR. This paper showcased the possibilities of FCA as a meta technique for categorizing the

literature on concept analysis with particular focus on the IR field. The intuitive visual interface of the concept lattices allowed for an in-depth exploration of the main research topics. Information retrieval is an important domain in which FCA was found to be an interesting instrument for representation of and navigation in large document collections and multiple IR systems resulted from this research. Also in query tuning and providing contextual answers to user queries, FCA was found to be a useful technique. In the future, we hope that this compendium may serve to guide both practitioners and researchers to new and improved avenues for FCA in the IR field.

Acknowledgements. Jonas Poelmans is aspirant of the “Fonds voor Wetenschappelijk Onderzoek – Vlaanderen” or “Research Foundation Flanders”.

References

1. Ahmad, I., Jang, T.S.: Old Fashion Text-Based Image Retrieval Using FCA. In: Proc. IEEE Int. Conf. Image Processing, ICIP-III, vol. 2, pp. 33–36 (2003)
2. Amato, G., Meghini, C.: Faceted Content-based Image Retrieval. In: Proc. 19th IEEE Int. Conf. on Database and Expert Systems Application, DEXA, pp. 402–406. (2008)
3. Bruno, M., Canfora, G., Penta, M.D., Scognamiglio, R.: An Approach to support Web Service Classification and Annotation. In: Proc. IEEE Int. Conf. on e-Technology, e-Commerce and e-Service, pp. 138–143 (2005)
4. Carpineto, C., Romano, G.: A lattice conceptual clustering system and its application to browsing retrieval. *Machine Learning* 24(2), 1–28 (1996b)
5. Carpineto, C., Romano, G.: *Concept data analysis: Theory and applications*. John Wiley & Sons (2004a)
6. Carpineto, C., Romano, G.: Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO. *J. of Universal Computing* 10(8), 985–1013 (2004b)
7. Carpineto, C., Romano, G.: Using Concept Lattices for Text Retrieval and Mining. In: Ganter, B., Stumme, G., Wille, R. (eds.) *ICFCA 2005*. LNCS (LNAI), vol. 3626, pp. 161–179. Springer, Heidelberg (2005)
8. Ceravolo, P., Gusmini, A., Leida, M., Cui, Z.: An FCA-based mapping generator. In: 12th IEEE int. Conf. on Emerging Technologies and Factory Automation, pp. 796–803 (2007)
9. Cigarrán, J.M., Gonzalo, J., Peñas, A., Verdejo, M.F.: Browsing Search Results via Formal Concept Analysis: Automatic Selection of Attributes. In: Eklund, P. (ed.) *ICFCA 2004*. LNCS (LNAI), vol. 2961, pp. 74–87. Springer, Heidelberg (2004)
10. Cigarrán, J.M., Peñas, A., Gonzalo, J., Verdejo, M.F.: Automatic Selection of Noun Phrases as Document Descriptors in an FCA-Based Information Retrieval System. In: Ganter, B., Godin, R. (eds.) *ICFCA 2005*. LNCS (LNAI), vol. 3403, pp. 49–63. Springer, Heidelberg (2005)
11. Cole, R., Eklund, P.: Browsing Semi-structured Web Texts Using Formal Concept Analysis. In: Delugach, H.S., Stumme, G. (eds.) *ICCS 2001*. LNCS (LNAI), vol. 2120, pp. 319–332. Springer, Heidelberg (2001)
12. Cole, R., Eklund, P., Stumme, G.: Document retrieval for e-mail search and discovery using Formal Concept Analysis. In: *Applied Artificial Intelligence*, vol. 17, pp. 257–280. Taylor & Francis (2003)
13. Cole, R.J.: *The management and visualization of document collections using Formal Concept Analysis*. Ph. D. Thesis, Griffith University (2000)

14. Ignatov, D.I., Kuznetsov, S.O.: Frequent Itemset Mining for Clustering Near Duplicate Web Documents. In: Rudolph, S., Dau, F., Kuznetsov, S.O. (eds.) ICCS 2009. LNCS (LNAI), vol. 5662, pp. 185–200. Springer, Heidelberg (2009)
15. Dau, F., Ducrou, J., Eklund, P.: Concept Similarity and Related Categories in Search-Sleuth. In: Eklund, P., Haemmerlé, O. (eds.) ICCS 2008. LNCS (LNAI), vol. 5113, pp. 255–268. Springer, Heidelberg (2008)
16. De Souza, K.X.S., Davis, J.: Using an Aligned Ontology to Process User Queries. In: Bussler, C.J., Fensel, D. (eds.) AIMSA 2004. LNCS (LNAI), vol. 3192, pp. 44–53. Springer, Heidelberg (2004)
17. Ducrou, J.: DVDSleuth: A Case Study in Applied Formal Concept Analysis for Navigating Web Catalogs. In: Priss, U., Polovina, S., Hill, R. (eds.) ICCS 2007. LNCS (LNAI), vol. 4604, pp. 496–500. Springer, Heidelberg (2007)
18. Ducrou, J., Vormbrock, B., Eklund, P.: FCA-Based Browsing and Searching of a Collection of Images. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) ICCS 2006. LNCS (LNAI), vol. 4068, pp. 203–214. Springer, Heidelberg (2006)
19. Ducrou, J., Eklund, P.W.: SearchSleuth: The Conceptual Neighborhood of an Web Query. In: CLA (2007b)
20. Ducrou, J., Wormuth, B., Eklund, P.: Dynamic Schema Navigation Using Formal Concept Analysis. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2005. LNCS, vol. 3589, pp. 398–407. Springer, Heidelberg (2005b)
21. Eklund, P., Ducrou, J.: Navigation and Annotation with Formal Concept Analysis. In: Richards, D., Kang, B.-H. (eds.) PKAW 2008. LNCS, vol. 5465, pp. 118–121. Springer, Heidelberg (2009)
22. Eklund, P., Ducrou, J., Brawn, P.: Concept Lattices for Information Visualization: Can Novices Read Line-Diagrams? In: Eklund, P. (ed.) ICFCA 2004. LNCS (LNAI), vol. 2961, pp. 57–73. Springer, Heidelberg (2004)
23. Eklund, P., Wille, R.: Semantology as Basis for Conceptual Knowledge Processing. In: Kuznetsov, S.O., Schmidt, S. (eds.) ICFCA 2007. LNCS (LNAI), vol. 4390, pp. 18–38. Springer, Heidelberg (2007)
24. Eklund, P., Wormuth, B.: Restructuring Help Systems Using Formal Concept Analysis. In: Ganter, B., Godin, R. (eds.) ICFCA 2005. LNCS (LNAI), vol. 3403, pp. 129–144. Springer, Heidelberg (2005)
25. Ganter, B., Wille, R.: Formal Concept Analysis. Mathematical foundations. Springer (1999)
26. Recio-García, J.A., Gómez-Martín, M.A., Díaz-Agudo, B., González-Calero, P.A.: Improving Annotation in the Semantic Web and Case Authoring in Textual CBR. In: Roth-Berghofer, T.R., Göker, M.H., Güvenir, H.A. (eds.) ECCBR 2006. LNCS (LNAI), vol. 4106, pp. 226–240. Springer, Heidelberg (2006)
27. Godin, R., Gecsei, J., Pichet, C.: Design of browsing interface for information retrieval. In: Belkin, N.J., et al. (eds.) Proc. GIR, pp. 32–39 (1989)
28. Godin, R., Missaoui, R., April, A.: Experimental comparison of navigation in a Galois lattice with conventional information retrieval methods. *Int. J. Man-Machine Studies* 38, 747–767 (1993)
29. Hachani, N., Ben Hassine, M.A., Chettaoui, H., et al.: Cooperative answering of fuzzy queries. *Journal of Computer Science and Technology* 24(4), 675–686 (2009)
30. Hitzler, P., Krötzsch, M.: Querying Formal Contexts with Answer Set Programs. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) ICCS 2006. LNCS (LNAI), vol. 4068, pp. 260–273. Springer, Heidelberg (2006)

31. Ignatov, D.I., Kuznetsov, S.O.: Frequent Itemset Mining for Clustering Near Duplicate Web Documents. In: Rudolph, S., Dau, F., Kuznetsov, S.O. (eds.) ICCS 2009. LNCS (LNAI), vol. 5662, pp. 185–200. Springer, Heidelberg (2009)
32. Kim, M., Compton, P.: Evolutionary Document Management and Retrieval for Specialised Domains on the Web. *Int. J. of Human Computer Studies* 60(2), 201–241 (2004)
33. Kim, M., Compton, P.: A Hybrid Browsing Mechanism Using Conceptual Scales. In: Hoffmann, A., Kang, B.-H., Richards, D., Tsumoto, S. (eds.) PKAW 2006. LNCS (LNAI), vol. 4303, pp. 132–143. Springer, Heidelberg (2006)
34. Koester, B.: Conceptual Knowledge Retrieval with FooCA: Improving Web Search Engine Results with Contexts and Concept Hierarchies. In: Perner, P. (ed.) ICDM 2006. LNCS (LNAI), vol. 4065, pp. 176–190. Springer, Heidelberg (2006)
35. Lakhal, L., Stumme, G.: Efficient Mining of Association Rules Based on Formal Concept Analysis. In: Ganter, B., Stumme, G., Wille, R. (eds.) ICFCFA 2005. LNCS (LNAI), vol. 3626, pp. 180–195. Springer, Heidelberg (2005)
36. Le Grand, B., Aufaure, M.A., Soto, M.: Semantic and Conceptual Context-Aware Information Retrieval. In: Damiani, E., Yetongnon, K., Chbeir, R., Dipanda, A. (eds.) SITIS 2006. LNCS, vol. 4879, pp. 247–258. Springer, Heidelberg (2009)
37. Liu, M., Shao, M., Zhang, W., Wu, C.: Reduction method for concept lattices based on rough set theory and its application. *Computers and Mathematics with Applications* 53, 1390–1410 (2007)
38. Lungley, D., Kruschwitz, U.: Automatically Maintained Domain Knowledge: Initial Findings. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) ECIR 2009. LNCS, vol. 5478, pp. 739–743. Springer, Heidelberg (2009)
39. Manning, C.D., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*. Cambridge University Press (2008)
40. Messai, N., Devignes, M.D., Napoli, A., Smail-Tabbone, M.: Extending Attribute Dependencies for Lattice-based Querying and Navigation. In: Eklund, P., Haemmerlé, O. (eds.) ICCS 2008. LNCS (LNAI), vol. 5113, pp. 189–202. Springer, Heidelberg (2008)
41. Messai, N., Devignes, M.-D., Napoli, A., Smail-Tabbone, M.: Querying a Bioinformatic Data Sources Registry with Concept Lattices. In: Dau, F., Mugnier, M.-L., Stumme, G. (eds.) ICCS 2005. LNCS (LNAI), vol. 3596, pp. 323–336. Springer, Heidelberg (2005)
42. Mili, H., Ah-Ki, E., Godin, R., Mcheick, H.: Another nail to the coffin of faceted controlled-vocabulary component classification and retrieval. *VCM SIGSOFT Software Engineering Notes* 22(3), 89–98 (1997)
43. Muangon, W., Intakosum, S.: Retrieving Design Patterns by Case-Based Reasoning and Formal Concept Analysis. In: 2nd Int. Conf. Comp. Sc. Inf. Technology, pp. 424–428 (2009)
44. Nafkha, I., Jaoua, A.: Using Formal Concept Analysis for Heterogeneous Information. In: Belohlavek, R.R., et al. (eds.) CLA, pp. 107–122 (2005)
45. Nauer, E., Toussaint, Y.: CreChainDo: An iterative and interactive Web information retrieval system based on lattices. *International Journal of General Systems* 38(4), 363–378 (2009)
46. Peng, D., Huang, S., Wang, X., Zhou, A.: Concept-Based Retrieval of Alternate Web Services. In: Zhou, L.-z., Ooi, B.-C., Meng, X. (eds.) DASFAA 2005. LNCS, vol. 3453, pp. 359–371. Springer, Heidelberg (2005a)
47. Peng, X., Zhao, W.: An Incremental and FCA-based Ontology Construction Method for Semantics-based Component Retrieval. In: 7th Int. Conf. on Quality Soft, pp. 309–315 (2007)

48. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: Formal Concept Analysis in Knowledge Discovery: A Survey. In: Croitoru, M., Ferré, S., Lukose, D. (eds.) ICCS 2010. LNCS, vol. 6208, pp. 139–153. Springer, Heidelberg (2010)
49. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: Concept Discovery Innovations in Law Enforcement: a Perspective. In: IEEE CINS Workshop (INCos), Greece (2010b)
50. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: Curbing domestic violence: Instantiating C-K theory with Formal Concept Analysis and Emergent Self Organizing Maps. *Intelligent Systems in Accounting, Finance and Management* 17(3-4), 167–191 (2010c)
51. Polaillon, G., Aufaure, M.A., Le Grand, B., Soto, M.: FCA for contextual semantic navigation and information retrieval in heterogeneous information systems. In: 8th IEEE Int. Workshop on Database and Expert Systems Applications, pp. 534–539 (2007)
52. Poshyvanyk, D., Marcus, A.: Combining Formal Concept Analysis with Information Retrieval for Concept Location in Source Code. In: Proc. IEEE Int. Conf. on Program Comprehension, pp. 37–48 (2007)
53. Priss, U.: Lattice-based Information Retrieval. *Knowledge Organization* 27(3), 132–142 (2000)
54. Priss, U.: Formal Concept Analysis in Information Science. In: Blaise, C. (ed.) Annual Review of Information Science and Technology, ASIST, vol. 40, pp. 521–543 (2006)
55. Spyrtos, N., Meghini, C.: Preference-Based Query Tuning Through Refinement/Enlargement in a Formal Context. In: Dix, J., Hegner, S.J. (eds.) FoIKS 2006. LNCS, vol. 3861, pp. 278–293. Springer, Heidelberg (2006)
56. Stojanovic, N.: On the query refinement in the ontology-based searching for information. *Information Systems* 30(7), 543–563 (2005)
57. Stojanovic, N.: On Using Query Neighborhood for Better Navigation through a Product Catalog: SMART Approach. In: IEEE Int. Conf. e-Tech., e-Com. and e-Service (2004)
58. Tane, J.: Using a Query-Based Multicontext for Knowledge Base Browsing. In: 3rd Int. Conf., ICFCA - Supplementary, Lens, France, pp. 62–78 (2005)
59. Tane, J., Cimiano, P., Hitzler, P.: Query-Based Multicontexts for Knowledge Base Browsing: An Evaluation. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) ICCS 2006. LNCS (LNAI), vol. 4068, pp. 413–426. Springer, Heidelberg (2006)
60. Tilley, T.: Tool Support for FCA. In: Eklund, P. (ed.) ICFCA 2004. LNCS (LNAI), vol. 2961, pp. 104–111. Springer, Heidelberg (2004)
61. Tilley, T., Eklund, P.: Citation analysis using Formal Concept Analysis: A case study in software engineering. In: 18th Int. Conf., DEXA, pp. 545–550 (2007)
62. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) *Ordered Sets*, pp. 445–470. Reidel, Dordrecht-Boston (1982)
63. Wille, R.: Methods of Conceptual Knowledge Processing. In: Missaoui, R., Schmidt, J. (eds.) ICFCA 2006. LNCS (LNAI), vol. 3874, pp. 1–29. Springer, Heidelberg (2006)
64. Zhang, Y., Feng, B., Xue, Y.: A New Search Results Clustering Algorithm based on Formal Concept Analysis. In: 5th Int. Conf. on FSKD, pp. 356–360 (2008)

Author Index

- Alves, Ronnie 92
Ammar, Reda A. 224
- Back, Barbro 52
Belo, Orlando 92
Ben Ahmed, Eya 194
Bichindaritz, Isabelle 9
Bobrowski, Leon 178
Burkovski, Andre 243
Bustillo, Andrés 67
Buza, Krisztian 38
- Dedene, Guido 258, 273
Dellmann, Frank 151
de Sousa, Jorge Freire 77
Dippon, Jürgen 243
- Eklund, Tomas 52
Elsayed, Samir A. Mohamed 224
Elzinga, Paul 258
- Ferreira, Carlos 77
Ferreira, Pedro 92
Fuellen, Georg 1
- Gama, João 77
Gargouri, Faiez 194
Gins, Geert 121
- Heidemann, Gunther 243
- Ignatov, Dmitry I. 273
Ivannikov, Andriy 103
- Jegoroff, Mikko 103
- Kärkkäinen, Tommi 103
Kim, Ieejoon 166
Kim, Ungmo 166
Kim, Younghee 166
Kirshners, Arnis 24
Klenk, Sebastian 243
Koch, Stephan 151
Kuznetsov, Sergei O. 258, 273
- Leja, Marcis 24
Lejri, Ons 113
Lim, Jiyeon 166
Liu, Hongyan 52
Lukaszuk, Tomasz 178
- Maudes, Jesús 67
Mendes-Moreira, João 77
Moreira-Matias, Luís 77
- Nabli, Ahlem 194
Nagy, Gabor I. 38
Neznanov, Alexei A. 258
- Paja, Wiesław 236
Park, Eunkyoung 166
Parshutin, Serge 24
Poelmans, Jonas 258, 273
- Rajasekaran, Sanguthevar 224
Reñones, Anfbal 67
Ribeiro, Joel 92
- Santos, Pedro 67
Schmidt, Rainer 1
Shimada, Kaoru 136
- Tagina, Moncef 113
- Van den Kerkhof, Pieter 121
Van Impe, Jan F.M. 121
Vanlaer, Jef 121
Viaene, Stijn 258, 273
Villa, Luisa F. 67
- Wang, Xia Li 209
Wang, Xiaochun 209
Weiss, Heike 1
Welcker, Laura 151
Wilkes, D. Mitch 209
Wrzesieñ, Mariusz 236
- Yao, Zhiyuan 52
Yoon, Jaeyeol 166